



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

KU LEUVEN

TOM RUBBRECHT

ON THE STATE ESTIMATION OF THREE PHASE MICRO-GRIDS WITH DISTRIBUTED PV GENERATORS

Master of Science thesis

Examiner: Prof. Enrique Acha
Examiner and topic approved by
the Faculty Council of Computing
and Electrical Engineering
on June 8th, 2016

ABSTRACT

TOM RUBBRECHT: On the State Estimation of Three Phase Micro-Grids with Distributed PV Generators

Tampere University of Technology

Master of Science thesis, 61 pages, 77 Appendix pages

July 2016

Master's Degree Programme in Electrical Energy Technology

Major: Electrical Engineering

Examiner: Prof. Enrique Acha

Keywords: Power Flow, State Estimation, VSC Modeling

This thesis investigates the application of three-phase State Estimation in Micro-Grids with distributed PV generators. In particular, it explores the use of three-phase Power Flows and State Estimation together with VSC modeling. The VSC may be used to provide voltage regulation at selected nodes of the micro-grid in which case it acts as a STATCOM or it may be used to provide the interface between the DC PV generator and the AC side of the micro-grid.

Small renewable energy plants are increasing their share of the overall energy resource. They are installed at the low-voltage end of the grid and this is blurring the otherwise classical concept of producers and consumers of electrical energy. More and more, consumers are able to produce electrical energy by installing PV cells at their premises and so they are becoming *prosumers*. This has resulted in the appearance of energy plants distributed all over the power grid. Hence, power flows might not be as straightforward to determine as previously, when producers and consumers were very clearly separated.

In order to ensure an effective control of the operation of the grid it is of paramount importance to have an effective knowledge of the actual state of the power network. Single-phase state estimators for power grids represent a classical concept in electric power systems theory but this is limited to positive-sequence representations of the power grid, i.e., three-phase power networks which exhibit a perfect balance. Needless to say that a major limitation of the positive-sequence network representation is its inapplicability to cases when the network unbalances are significant and cannot be ignored. In real life the load is not always equally spread between the three

phases; no matter how hard engineers try to achieve a reasonable balance, there is always a degree of unbalance which ought to be studied because it may impair the operation of the grid and increase its power losses. This can only be assessed by representing the three-phase power grid on its natural coordinate system, which is the *abc* frame-of-reference.

This thesis presents a three-phase modeling approach to power flows and state estimation of balanced and unbalanced power networks. A second contribution is the modeling of a three-phase VSC that has been derived from an existing single-phase VSC. The model is then extended to incorporate the model of PV generator. Because of the flexibility with which the code has been developed, the incorporation of multiple STATCOMs and PV generators may be simulated to be a realistic part of the power grid, enabling the distributed concept and the micro-grid concept to be brought together.

PREFACE

This Master of Science thesis was written at the department of Electrical Engineering at Tampere University of Technology. My supervisor and examiner was Professor Enrique Acha.

First of all, I would like to thank Professor Enrique Acha for providing and guiding me through this interesting thesis topic. When I arrived at TUT I was unaware about the contents of the thesis subject. During the semester I learned a lot, which would not have been possible for me without the support and guidance of my supervisor.

I would like to thank everyone who helped and supported me during my thesis, all my friends at home and all the new friends I met during my stay in Tampere, with whom I shared the most memorable moments during the exchange period.

I also want to thank my family for their support and the opportunity they gave me to study abroad.

And last but not least, thanks to all the people from my home university KU Leuven, especially Mr. Philip Keersebilck and Mrs. Christine Van Laere, and the host university Tampere University of Technology, who made my Erasmus in Tampere possible. Thank you TUT for the great hospitality and all the possibilities you give the students. Thank you KU Leuven for giving me the opportunity to make my master thesis in Tampere.

Tampere, 29.7.2016

Tom Rubbrecht

TABLE OF CONTENTS

1. Introduction	1
1.1 Theoretical background	2
1.2 Thesis outline	2
2. Power flows	3
2.1 Three-phase power flows	4
2.1.1 Power flow equations	5
2.1.2 Iterative solution methods	6
2.1.3 Power flow solution	9
2.2 Three phase power flows with VSC	11
2.2.1 VSC without control	11
2.2.2 VSC with voltage control	11
2.2.3 VSC with power flow control	12
2.2.4 VSC with combined voltage and power flow control	12
2.3 Test cases	13
2.3.1 Power flow with cable parameters	13
2.3.2 Power flows with VSC	16
3. State estimation	21
3.1 Three phase state estimation algorithm	22
3.1.1 Errors	23
3.1.2 Jacobian matrix H_x	25
3.1.3 Bad data	28
3.2 Three phase state estimation with VSC	31
3.2.1 VSC without control	31
3.2.2 VSC with voltage control	31
3.2.3 VSC with power flow control	31

3.2.4	VSC with combined voltage and power flow control	32
3.3	Test cases	32
3.3.1	Checking the three phase state estimation program	33
3.3.2	State estimation with VSC	35
4.	Micro-grids	38
4.1	VSC model for power flows	38
4.1.1	Three phase power flow model	39
4.1.2	The nodal power equations	40
4.1.3	The mismatch power flow equations	42
4.2	Power flow model for a DC-DC Converter	43
4.2.1	Test case	44
4.3	Cable modeling	47
4.3.1	Series impedance matrix of power cables	48
4.3.2	Shunt admittance matrix of power cables	49
4.4	Micro-grid: test case	50
4.4.1	Power flow solution	53
4.4.2	State estimation solution	57
5.	Summary	60
5.1	Recommendations for future work	60
	Bibliography	62
	APPENDIX A. Power Flow Solutions of AC/DC Micro-grid Structures	63
	APPENDIX B. The Power Flow Program	70
	APPENDIX C. The State Estimation Program	100

LIST OF FIGURES

2.1	Three-phase equivalent transmission line [1]	4
2.2	Graphical representation of the general Newton-Rapson algorithm [2]	7
2.3	Power Flow algorithm flowchart	10
2.4	Configuration of the network for Test Case 2.1	13
2.5	Test Case 2.1: Cable installation configuration in underground ducts	14
2.6	Configuration of the network for Test Case 2.2	16
3.1	Standard Normal probability density function $p(z)$ [16]	24
3.2	χ^2 -distribution for $m - n$ degrees of freedom and a confidence interval of α	29
3.3	State Estimation algorithm flowchart	30
3.4	Configuration of the network for Test Case 3.1	33
4.1	Three phase Invertor principle scheme	39
4.2	Three phase VSC power flow model	39
4.3	Graphical representation of the power flow mismatch function	43
4.4	Principle scheme of the buck-boost Converter	43
4.5	Configuration of the network for Test Case 4.1	44
4.6	Configuration of the network for Test Case 4.2	52
4.7	Directions of the power flow for Test Case 4.2	57

LIST OF TABLES

2.1	Classification of buses and variables	4
2.2	Test Case 2.1: Voltage Magnitudes	15
2.3	Test Case 2.1: Voltage Angles	15
2.4	Test Case 2.1: Forward Power Flows	15
2.5	Test Case 2.1: Reversed Power Flows	15
2.6	Test Case 2.1: Power losses	15
2.7	Test Case 2.2a: Voltage Magnitudes	16
2.8	Test Case 2.2a: Voltage Angles	17
2.9	Test Case 2.2a: Forward Power Flows	17
2.10	Test Case 2.2a: Reversed Power Flows	17
2.11	Test Case 2.2a: Power losses	17
2.12	Test Case 2.2b: Voltage Magnitudes	18
2.13	Test Case 2.2b: Voltage Angles	18
2.14	Test Case 2.2b: Amplitude modulation ratio	18
2.15	Test Case 2.2b: Forward Power Flows	18
2.16	Test Case 2.2b: Reversed Power Flows	18
2.17	Test Case 2.2b: Power losses	18
2.18	Test Case 2.2c: Voltage Magnitudes	19
2.19	Test Case 2.2c: Voltage Angles	19

2.20 Test Case 2.2c: Amplitude modulation ratio	19
2.21 Test Case 2.2c: Phase shifting angle	19
2.22 Test Case 2.2c: Equivalent susceptance	19
2.23 Test Case 2.2c: Forward Power Flows	20
2.24 Test Case 2.2c: Reversed Power Flows	20
2.25 Test Case 2.2c: Power losses	20
3.1 Test Case 3.1: Measurement Data	34
3.2 Test Case 3.1: Estimated voltage magnitudes	34
3.3 Test Case 3.1: Estimated voltage angles	35
3.4 Test Case 3.1: Results in positive sequence	35
3.5 Test Case 3.2a: Estimated voltage magnitudes	36
3.6 Test Case 3.2a: Estimated voltage angles	36
3.7 Test Case 3.2b: Estimated voltage magnitudes	36
3.8 Test Case 3.2b: Estimated voltage angles	36
3.9 Test Case 3.2b: Estimated amplitude modulation ratio	36
3.10 Test Case 3.2c: Estimated voltage magnitudes	37
3.11 Test Case 3.2c: Estimated voltage angles	37
3.12 Test Case 3.2c: Estimated amplitude modulation ratio	37
3.13 Test Case 3.2c: Estimated phase shifting angle	37
3.14 Test Case 3.2c: Estimated equivalent susceptance	37
4.1 Test Case 4.1a: Voltage Magnitudes	45

4.2 Test Case 4.1a: Voltage Angles	45
4.3 Test Case 4.1a: Amplitude modulation ratio	45
4.4 Test Case 4.1a: Phase shifting angle	45
4.5 Test Case 4.1a: Forward Power Flows	46
4.6 Test Case 4.1a: Reversed Power Flows	46
4.7 Test Case 4.1a: Power losses	46
4.8 Test Case 4.1b: Estimated voltage magnitudes	46
4.9 Test Case 4.1b: Estimated voltage angles	47
4.10 Test Case 4.1b: Estimated amplitude modulation ratio	47
4.11 Test Case 4.1b: Estimated phase shifting angle	47
4.12 Test Case 4.2: Transformer Parameters	51
4.13 Test Case 4.2: Cable Parameters	51
4.14 Test Case 4.2: Sources and Loads	52
4.15 Test Case 4.2a: Voltage Magnitudes	53
4.16 Test Case 4.2a: Voltage Angles	54
4.17 Test Case 4.2a: Forward Power Flows	55
4.18 Test Case 4.2a: Reversed Power Flows	55
4.19 Test Case 4.2a: Power losses	56
4.20 Test Case 4.2b: Estimated voltage magnitudes	57
4.21 Test Case 4.2b: Estimated voltage angles	59

LIST OF ABBREVIATIONS AND SYMBOLS

Abbreviations

AC	Alternating Current
DC	Direct Current
HVDC	High Voltage Direct Current
pdf	Probability Density Function
PQ-bus	Load bus
p.u.	Per Unit
PV	Photo-Voltaic
PV-bus	Voltage Controlled bus or Generator bus
SCADA	Supervisory Control And Data Acquisition
STATCOM	Static Synchronous Compensator
TL	Transmission Line
VA	Voltage Angle
VM	Voltage Magnitude
VSC	Voltage Source Converter

Symbols

B	Susceptance
B'	Line-charging susceptance
C	Capacitor
e	Error
E	Voltage
f	Objective function
G	Conductance
G	Gain matrix
h	Jacobian entry
H_x	Jacobian matrix in State Estimation
i	iteration
I	Current
J	Jacobian matrix in Power Flow
L	Inductor

m	Number of measurements
m_ρ	amplitude modulation ratio of phase ρ
n	Number of state variables
$p(z)$	Probability density function of z
P	Active power
P_x	Measured active power
P_{xy}	Measured active power flow between bus x and y
PQ	Active and reactive power flow in complex notation
$PQlosses$	Active and reactive power losses
Q	Reactive power
Q_x	Measured reactive power
Q_{xy}	Measured reactive power flow between bus x and y
R	Resistance
S	Apparent power
V	Voltage
w	Weighting factor
x	State variable
X	Reactance
Y	Admittance
z	Measurement
Z	Impedance
α	Confidence interval
δ	Duty cycle
Δx	Mismatch of x
θ	Voltage angle
μ	Expected value
ρ	Phase identification
σ	Standard deviation
σ^2	Variance
ϕ	Phase shifting angle

1. INTRODUCTION

Electric power grids have been changing drastically in the last years, more and more traditional power plants are replaced by newer, more environmental friendly power plants. Especially solar panels have been extremely popular by consumers. These consumers saw the economical or environmental benefit to produce power, lower there energy bill and inject power into the grid. However our traditional grids were not designed for distributed power injections. New concepts like micro-grids and smart grids were introduced. What are there goals? To incorporate new technologies and control methods which would enable more economical and energy friendly alternatives to traditional electrical power grids. These goals are rather challenging to achieve: the distributed power generation blurs out the traditional concepts of producer and consumers and introduces a new term, *prosumers*. Hence, power flows might not be as straightforward to determine as previously, when producers and consumers were very clearly separated.

To control an electrical power grid, it is critical to have a realistic knowledge of the actual state of the system. Power flow and state estimation programs need to be adapted to incorporate the distributed generation philosophy. Single-phase power flow and state estimation programs are well known. These programs provide good results, but their biggest disadvantage is their use of the positive sequence representation concept. The coherent implementation of the positive sequence representation is quite straightforward to achieve reasonable accurate results. Nevertheless in case of significant network unbalance, the positive sequence representation is unable to give an accurate assessment of the power grid.

It goes without saying that network unbalances should be avoided at the design stage. However, the load is not always spread equally between the three phases; no matter how hard engineers try to achieve a reasonable balance, there is always a degree of unbalance. In many cases, single or two phase loads are serviced. Single-phase loads are an integral part of the electrical power grid and they ought to

be studied as such because they impact the operation of the grid more than an equivalent three-phase load. This can only be assessed by representing the three-phase power grid in the *abc* frame-of-reference. This is the motivation behind this thesis, to present a three-phase modeling approach with emphasis on power flows and state estimation. A key contribution in this thesis is the model of a three phase Voltage Source Converter (VSC). The VSC model can be extended to implement the model of a photo-voltaic (PV) generator, enabling the possibility to assess power networks with distributed PV generation.

1.1 Theoretical background

In order to fully understand this thesis report a basic knowledge of power systems is needed. This requires the knowledge of the *abc* frame-of-reference, the theory of symmetrical components and the per-unit (*p.u.*) system.

Both power flow and state estimation algorithms rely heavily on linear algebra. A good understanding of matrix analysis is highly recommended.

1.2 Thesis outline

This thesis consists of five chapters. Chapter 2 explains the three phase power flow algorithm and provides the techniques to implement a three phase VSC model in the power flow program. After the implementation of the VSC in the power flow program, several control options of the VSC will be tested using the MATLAB scripting environment. Chapter 3 offers the basic strategies of three-phase state estimation and the implementation of the VSC model. A three-phase state estimation program is built using the techniques provided in this chapter. The test cases in chapter 3 are used to verify the programmed code. Once the program is verified, a larger test network can be provided to the program, which will be discussed in chapter 4. Chapter 4 deals with the three-phase VSC power flow model that has been made to fit into the three-phase programs as well as the cable modeling used for the test networks. The thesis is summarized in chapter 5.

2. POWER FLOWS

The power flow analysis involves the steady state solution of a power system network for a specific generation and loading. The power flow, voltage magnitudes and voltage angles can be determined. Once all the power flow and injected powers are known the total power losses in the network can be found. These studies are important for planning and future expansion of the power system and can be used to optimize the system.

Knowing the power flow is essential when operating a grid. The demand of power changes constantly. In order to get more control, it's of utmost importance to know which transmission lines can be used to deliver extra power without any risk of damage or failure [5]. In the liberated energy market, power can also be traded with other companies or country's. Different scenarios can be simulated in order to get the optimal power flow, with the least losses and thus the most cost-effective way to distribute energy.

Associated with each bus are four quantities per phase that fully determines the state of the grid: the active and reactive power, the voltage magnitude and voltage angle. At most there are two quantities known per phase. It must be considered how the unknowns can be reduced to agree with the number of available equations. For this reason three types of buses will be defined: [9, 10]

1. *The Slack bus* acts as a reference for all the calculations and is a special case of bus 2, the voltage controlled bus. The voltage angle of the slack bus serves as a reference for the angles of all other buses. The initial angle of the slack bus is not important because only the difference in voltage angle defines the calculated values, namely, the active and reactive power. Common practice sets the voltage angle of the slack bus to 0° , 120° and 240° for respectively phase a , b and c . The voltage magnitudes are also specified and are normally set to 1 *p.u.*

2. *The Voltage controlled bus* is any bus of the system where the voltage magnitude is kept constant. Each generator bus with generation control is capable of controlling the voltage magnitude at the node, and is termed PV-bus. The short notation form already shows which quantities are known at the bus: the active power injected at the node and the voltage magnitude. The reactive power and the voltage angle need to be calculated.
3. *The load bus* or PQ-bus is a type of bus which contains no generator. The active and reactive powers consumed at the node are known. The voltage magnitude and voltage angle need to be calculated.

The following table summarizes the different types of buses and their associated variables.

Table 2.1 Classification of buses and variables

Type of bus	known information	calculated information
(1) Slack-bus	V, θ	P, Q
(2) PV-bus	V, P	Q, θ
(3) PQ-bus	P, Q	V, θ

2.1 Three-phase power flows

To derive the basic power flow equations, suitable relationships between the bus current and the bus voltage must be established. Starting with a three phase representation of an equivalent transmission line connected between bus k and bus m , given in figure 2.1, the basic relationship can be found by resorting to Ohm's law,

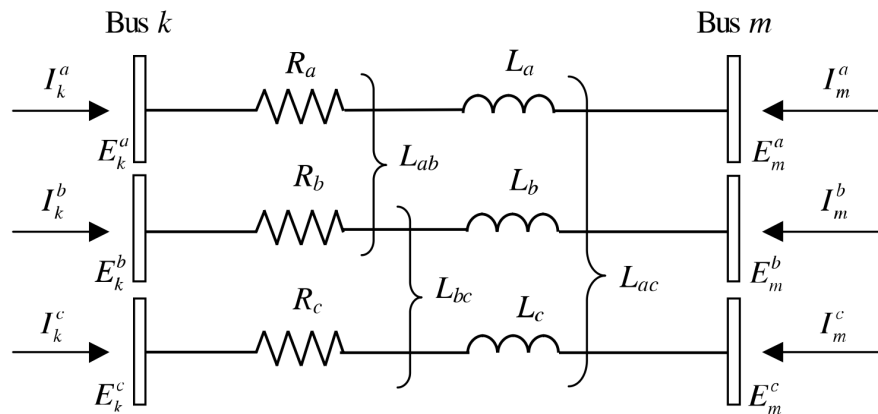


Figure 2.1 Three-phase equivalent transmission line [1]

$$\begin{bmatrix} I_k^{abc} \\ I_m^{abc} \end{bmatrix} = \begin{bmatrix} Y_{kk}^{abc} & Y_{km}^{abc} \\ Y_{mk}^{abc} & Y_{mm}^{abc} \end{bmatrix} \cdot \begin{bmatrix} E_k^{abc} \\ E_m^{abc} \end{bmatrix} \quad (2.1)$$

where

$$Y_{kk}^{abc} = \begin{bmatrix} \overline{Y}_{km}^{aa-g} & \overline{Y}_{km}^{ab-g} & \overline{Y}_{km}^{ac-g} \\ \overline{Y}_{km}^{ba-g} & \overline{Y}_{km}^{bb-g} & \overline{Y}_{km}^{bc-g} \\ \overline{Y}_{km}^{ca-g} & \overline{Y}_{km}^{cb-g} & \overline{Y}_{km}^{cc-g} \end{bmatrix}$$

$$Y_{kk}^{abc} = Y_{mm}^{abc} = -Y_{km}^{abc} = -Y_{mk}^{abc}$$

$$I_k^{abc} = \begin{bmatrix} \overline{I}_k^a & \overline{I}_k^b & \overline{I}_k^c \end{bmatrix}^T$$

$$I_m^{abc} = \begin{bmatrix} \overline{I}_m^a & \overline{I}_m^b & \overline{I}_m^c \end{bmatrix}^T$$

$$E_k^{abc} = \begin{bmatrix} \overline{E}_k^a & \overline{E}_k^b & \overline{E}_k^c \end{bmatrix}^T$$

$$E_m^{abc} = \begin{bmatrix} \overline{E}_m^a & \overline{E}_m^b & \overline{E}_m^c \end{bmatrix}^T$$

The admittance matrix Y_{kk}^{abc} is the self admittance matrix of bus k ; its diagonal elements represent the self-admittance in each phase. All non-diagonal elements are the mutual admittances between any two phases. All elements include the contribution of the ground return, represented by the superscript g [1]. The admittance matrices Y_{mm}^{abc} , Y_{km}^{abc} and Y_{mk}^{abc} have similar descriptions.

2.1.1 Power flow equations

In each phase the active and reactive powers can be written in complex form:

$$\overline{S}_k = P_k + j \cdot Q_k = \overline{V}_k \overline{I}_k^* \quad (2.2)$$

where \overline{I}_k is the sum of all the current injections into bus k at a given phase:

$$\overline{I}_k = \sum_{m=1}^n \overline{I}_{km}$$

Invoking Ohm's law: $\bar{I} = \bar{Y} \cdot \bar{V}$, eq.(2.2) can be written down for a three phase network as follows:

$$\bar{S}_k^{abc} = P_k^{abc} + j \cdot Q_k^{abc} = \bar{V}_k^{abc} \sum_{m=1}^n \bar{I}_{km}^{abc*} = \bar{V}_k^{abc} \sum_{m=1}^n \bar{Y}_{km}^{abc*} \cdot \bar{V}_m^{abc*} \quad (2.3)$$

After some complex algebra, the real and imaginary parts can be extracted:

$$P_k^\rho = V_k^\rho \sum_{i=k,m} \sum_{j=a,b,c} V_i^j G_{ki}^{\rho j} \cos(\theta_k^\rho - \theta_i^j) + B_{ki}^{\rho j} \sin(\theta_k^\rho - \theta_i^j) \quad (2.4)$$

$$Q_k^\rho = V_k^\rho \sum_{i=k,m} \sum_{j=a,b,c} V_i^j G_{ki}^{\rho j} \sin(\theta_k^\rho - \theta_i^j) - B_{ki}^{\rho j} \cos(\theta_k^\rho - \theta_i^j) \quad (2.5)$$

The superscripts ρ and j are used to denote the phases a, b, c . Subscript i is used to denote the bus. Eq.(2.4) and eq.(2.5) represent the active and reactive power injections at bus k in phase ρ . Similar equations can be derived for bus m . Note that these power injection equations are general, with the subscript k comprising nodes 1 to n .

2.1.2 Iterative solution methods

The injected power equations are non linear and must be solved using an iterative method. The method of choice in this master thesis is the Newton-Raphson algorithm, owing to its strong convergence characteristics.

General Newton-Raphson method

The Newton-Raphson algorithm can find the roots of a function by repetitively making better approximations to the roots. As in any iterative method, we ought to provide an initial guess for the values of the state variables [15]. At the onset the function value $f(x)$ is calculated using the initial guess, at a point which corresponds to a linearized region of the function $[x, f(x)]$. The roots of the linearized equations will be calculated and the initial values will be updated using the mismatch Δx , thus completing one iteration. In the iteration the function value will be recalculated using the updated set of variables. If this function value is bigger than the tolerance, the next iteration will start. On the other hand, if the function is smaller

than the tolerance, the iterative procedure stops and the solution is reached. The tolerance level is set to 1E-12 for convergence.

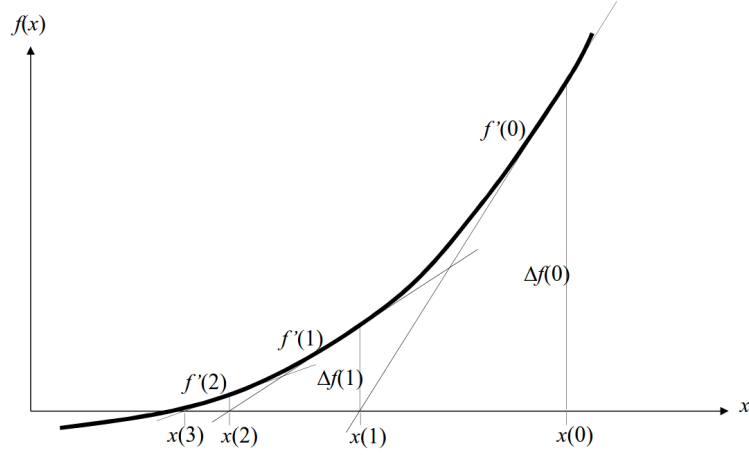


Figure 2.2 Graphical representation of the general Newton-Raphson algorithm [2]

Newton-Raphson method for three phase power flow

In the absence of a better estimate, the so called *flat start* is recommended since it requires the least number of iterations. [8] In a flat start all nodal voltage magnitudes will be set to 1 *p.u.* and all phase angles will be set to 0°, 120° or 240°, respectively phase *a*, *b* and *c*, except in the voltage controlled nodes where the voltage magnitude has been set at a value different from the 1 *p.u.*

At each iteration *i*, eq.(2.4) and eq.(2.5) will be linearized:

$$\Delta P_k = \frac{\partial P_k}{\partial \theta_k^a} \Delta \theta_k^a + \dots + \frac{\partial P_k}{\partial \theta_m^c} \Delta \theta_m^c + \frac{\partial P_k}{\partial V_k^a} \Delta V_k^a + \dots + \frac{\partial P_k}{\partial V_m^c} \Delta V_m^c \quad (2.6)$$

$$\Delta Q_k = \frac{\partial Q_k}{\partial \theta_k^a} \Delta \theta_k^a + \dots + \frac{\partial Q_k}{\partial \theta_m^c} \Delta \theta_m^c + \frac{\partial Q_k}{\partial V_k^a} \Delta V_k^a + \dots + \frac{\partial Q_k}{\partial V_m^c} \Delta V_m^c \quad (2.7)$$

The linearization takes into account all the buses in the network except the slack bus. The slack bus is used as a reference for all the voltage magnitudes and voltage angles. Since the values of the slack bus do not change, the partial derivatives of the slack bus are zero.

In matrix notation, the linearized equations, corresponding to iteration *i*, are given

below:

$$\begin{bmatrix} \Delta P_l^\rho \\ \Delta Q_l^\rho \end{bmatrix}^{(i)} = \underbrace{\begin{bmatrix} \frac{\partial P_l^\rho}{\partial \theta_j^\rho} & \frac{\partial P_l^\rho}{\partial V_j^\rho} V_j^\rho \\ \frac{\partial Q_l^\rho}{\partial \theta_j^\rho} & \frac{\partial Q_l^\rho}{\partial V_j^\rho} V_j^\rho \end{bmatrix}}_{\text{Jacobian}}^{(i)} \cdot \begin{bmatrix} \Delta \theta_j^\rho \\ \Delta V_j^\rho / V_j^\rho \end{bmatrix}^{(i)} \quad (2.8)$$

where $l = k, m$; $j = k, m$.

The vector terms take the following form:

$$\begin{aligned} \Delta P_l^\rho &= \begin{bmatrix} \Delta P_k^a & \Delta P_k^b & \Delta P_k^c & \Delta P_m^a & \Delta P_m^b & \Delta P_m^c \end{bmatrix}^T \\ \Delta Q_l^\rho &= \begin{bmatrix} \Delta Q_k^a & \Delta Q_k^b & \Delta Q_k^c & \Delta Q_m^a & \Delta Q_m^b & \Delta Q_m^c \end{bmatrix}^T \\ \Delta \theta_j^\rho &= \begin{bmatrix} \Delta \theta_k^a & \Delta \theta_k^b & \Delta \theta_k^c & \Delta \theta_m^a & \Delta \theta_m^b & \Delta \theta_m^c \end{bmatrix}^T \\ \frac{\Delta V_j^\rho}{V_j^\rho} &= \begin{bmatrix} \frac{\Delta V_k^a}{V_k^a} & \frac{\Delta V_k^b}{V_k^b} & \frac{\Delta V_k^c}{V_k^c} & \frac{\Delta V_m^a}{V_m^a} & \frac{\Delta V_m^b}{V_m^b} & \frac{\Delta V_m^c}{V_m^c} \end{bmatrix}^T \end{aligned}$$

The matrix of the first order partial derivatives is called Jacobian matrix. The Jacobian needs to be recalculated at each iteration.

Equations (2.4) - (2.8) have been derived for the three-phase transmission line in figure (2.1). However, they can easily be extended to apply to a whole network.

Equation (2.8) will be solved for the mismatch vectors. After each iteration the state variables are updated using the state variables increments. If the function value is smaller than the specified tolerance ε the convergence criterion is met. If the tolerance has not yet been reached, then the iterative process continues. Normally the tolerance criterion is set at $\varepsilon = 1\text{E-}12$.

The mismatch power vectors, at iteration i , may be defined as:

$$\Delta P^{(i)} = (P^{gen} - P^{load}) - P^{calc,(i)} = P^{net} - P^{calc,(i)} \quad (2.9)$$

$$\Delta Q^{(i)} = (Q^{gen} - Q^{load}) - Q^{calc,(i)} = Q^{net} - Q^{calc,(i)} \quad (2.10)$$

P^{net} and Q^{net} represent the net powers at each bus, sometimes they are called the scheduled powers. These scheduled powers are known and do not change throughout the iterations. This is in contrast to the calculated powers. If $\Delta P^{(i)}$ and $\Delta Q^{(i)}$ are equal to zero with a deviation of ε , then the solution has converged.

After solving the linearized set of equations, at each iteration, the angle and voltage

vectors are updated:

$$\theta^{(i)} = \theta^{(i-1)} + \Delta\theta^{(i)} \quad (2.11)$$

$$V^{(i)} = V^{(i-1)} + \Delta V^{(i)} V^{(i-1)} \quad (2.12)$$

2.1.3 Power flow solution

Once the voltage solution is reached by iteration, the power flows can be calculated. Using the voltage magnitude and the voltage angle solution, the nodal power injections are calculated using eq.(2.4) and eq.(2.5). The power flows between two connected buses in phase ρ can be calculated in a similar way. The power losses at a given transmission line (TL) can be calculated by adding up the power flows entering both nodes.

For example the calculated power flows and power losses between bus k and m in phase ρ , can be calculated as follows:

$$\begin{aligned} P_{km}^\rho &= V_k^\rho V_m^\rho G_{km}^\rho \cos(\theta_k^\rho - \theta_m^\rho) + B_{km}^\rho \sin(\theta_k^\rho - \theta_m^\rho) \\ Q_{km}^\rho &= V_k^\rho V_m^\rho G_{km}^\rho \sin(\theta_k^\rho - \theta_m^\rho) - B_{km}^\rho \cos(\theta_k^\rho - \theta_m^\rho) \end{aligned} \quad (2.13)$$

$$\begin{aligned} P_{km-loss}^\rho &= P_{km}^\rho + P_{mk}^\rho \\ Q_{km-loss}^\rho &= Q_{km}^\rho + Q_{mk}^\rho \end{aligned} \quad (2.14)$$

A flowchart for the power flow algorithm using the Newton-Raphson method is shown in fig.(2.3).

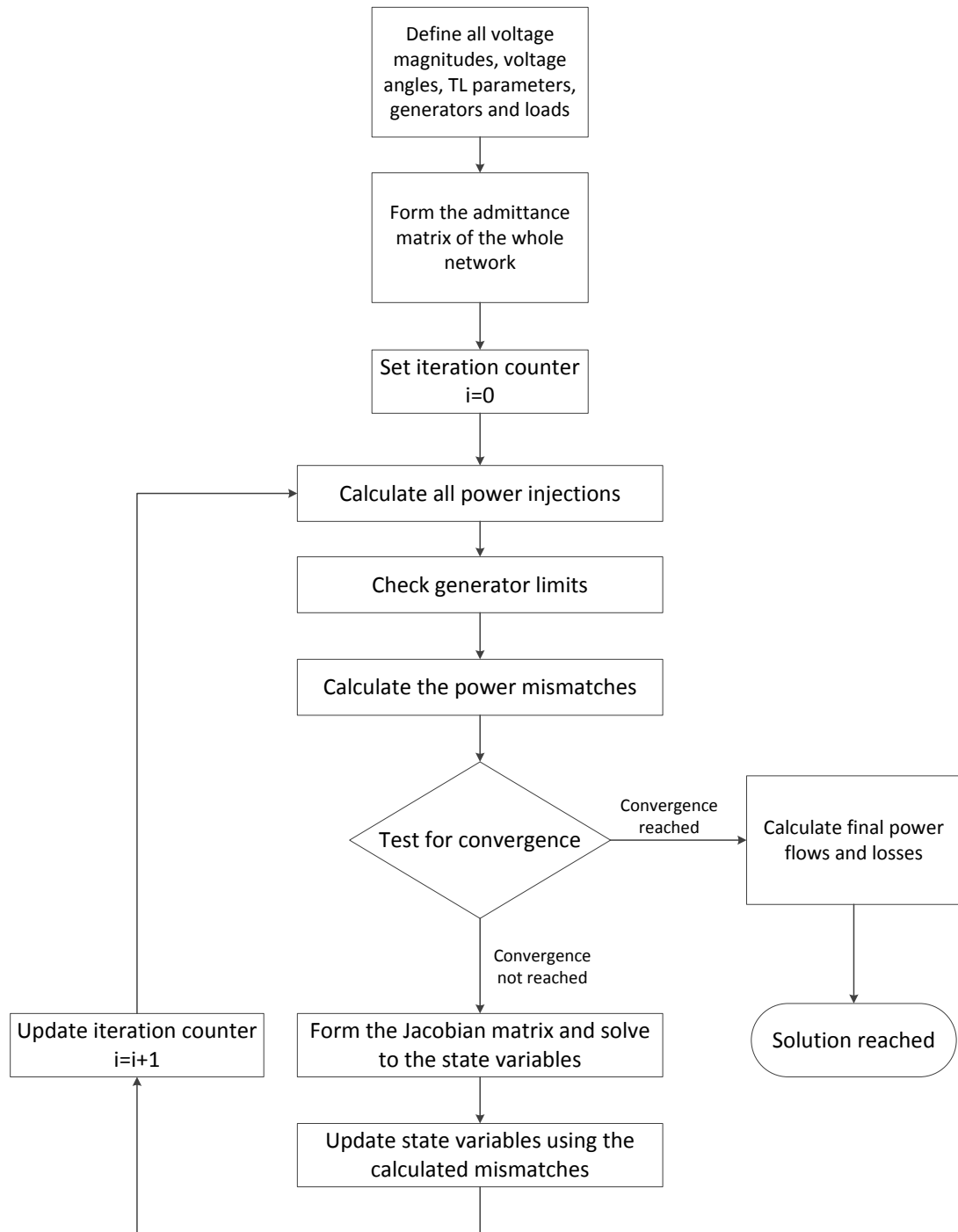


Figure 2.3 Power Flow algorithm flowchart

2.2 Three phase power flows with VSC

The next step in the power flow research project is to implement the model of a Voltage Source Converter (VSC). This model can be extended to incorporate the model of a Photo-Voltaic (PV) generator, an issue to be discussed in Chapter 4.

The VSC impacts the Jacobian matrix of the power flow formulation, not least because of its three different control modes. The simplest option is to have no control. A second option includes voltage control on either the AC or DC sides. Another control option relates to the power flow regulation at one end of the VSC. The combined voltage power flow control is one further option.

2.2.1 VSC without control

When implementing a VSC without control; no control in either side nor power flow regulations, the modifications in the Jacobian matrix are minimal. The Jacobian has the same structure as in eq.(2.8), corresponding to the normal power flow. For the implementation of the VSC, its contribution needs to be added to the injected active and reactive powers at the nodes where the VSC is connected to. The total calculated powers at the node will be the sum of the normal network calculated powers and the VSC calculated powers, which also need to be included in the power mismatches:

$$\begin{aligned}\Delta P_k^{(i)} &= P_k^{net} - \sum P_k^{calc,(i)} = P_k^{net} - \left(P_k^{calc,(i)} + P_{k,VSC}^{calc,(i)} \right) \\ \Delta Q_k^{(i)} &= Q_k^{net} - \sum Q_k^{calc,(i)} = Q_k^{net} - \left(Q_k^{calc,(i)} + Q_{k,VSC}^{calc,(i)} \right)\end{aligned}$$

2.2.2 VSC with voltage control

In the case of a voltage controlled VSC, the node which is undergoing control does not require the entries corresponding to the voltage magnitude. The voltage is kept constant and it is not a variable anymore. The controlled voltage only depends on the voltage on the other side of the VSC and its amplitude modulation ratio m_ρ . Hence the entries corresponding to the nodal voltage magnitude are replaced by the entries at the amplitude modulation ratio; the size of the Jacobian does not change.

In the test case 2.2b (see section 2.3.2), where the voltage is controlled at the sending end, the linearized equations of bus 2 looks as follow:

$$\begin{aligned}\Delta P_2^\rho &= \frac{\partial P_2^\rho}{\partial \theta_2^\rho} \Delta \theta_2^\rho + \frac{\partial P_2^\rho}{\partial \theta_3^\rho} \Delta \theta_3^\rho + \frac{\partial P_2}{\partial m_\rho} \Delta m_\rho + \frac{\partial P_2}{\partial V_3^\rho} \Delta V_3^\rho \\ \Delta Q_2^\rho &= \frac{\partial Q_2^\rho}{\partial \theta_2^\rho} \Delta \theta_2^\rho + \frac{\partial Q_2^\rho}{\partial \theta_3^\rho} \Delta \theta_3^\rho + \frac{\partial Q_2}{\partial m_\rho} \Delta m_\rho + \frac{\partial Q_2}{\partial V_3^\rho} \Delta V_3^\rho\end{aligned}$$

where V_3 is the voltage at the other side of the VSC (DC-side).

2.2.3 VSC with power flow control

When power flow control is applied to the VSC, two extra state variables per phase will be added to the system of equations: the phase shifting angle ϕ_ρ and the equivalent susceptance B_{eq}^ρ . Since the number of equations needs to match the number of state variables, two extra equations per phase are introduced to fulfill this requirement. The mismatch power flow equations, as explained in section 4.1.3, serve such a purpose. The mismatch power flow equations express the differences that exist between the regulated power flow and the calculated power flow in the VSC. Eq. (2.8) becomes the following one:

$$\begin{bmatrix} \Delta P_l^\rho \\ \Delta Q_l^\rho \\ \Delta P_{0-Vr}^\rho \\ \Delta Q_{0-Vr}^\rho \end{bmatrix}^{(i)} = \begin{bmatrix} \frac{\partial P_l^\rho}{\partial \theta_j^\rho} & \frac{\partial P_l^\rho}{\partial V_j^\rho} V_j^\rho & \frac{\partial P_l^\rho}{\partial \phi_\rho} & \frac{\partial P_l^\rho}{\partial B_{eq}^\rho} \\ \frac{\partial Q_l^\rho}{\partial \theta_j^\rho} & \frac{\partial Q_l^\rho}{\partial V_j^\rho} V_j^\rho & \frac{\partial Q_l^\rho}{\partial \phi_\rho} & \frac{\partial Q_l^\rho}{\partial B_{eq}^\rho} \\ \frac{\partial P_{0-Vr}^\rho}{\partial \theta_j^\rho} & \frac{\partial P_{0-Vr}^\rho}{\partial V_j^\rho} V_j^\rho & \frac{\partial P_{0-Vr}^\rho}{\partial \phi_\rho} & \frac{\partial P_{0-Vr}^\rho}{\partial B_{eq}^\rho} \\ \frac{\partial Q_{0-Vr}^\rho}{\partial \theta_j^\rho} & \frac{\partial Q_{0-Vr}^\rho}{\partial V_j^\rho} V_j^\rho & \frac{\partial Q_{0-Vr}^\rho}{\partial \phi_\rho} & \frac{\partial Q_{0-Vr}^\rho}{\partial B_{eq}^\rho} \end{bmatrix}^{(i)} \cdot \begin{bmatrix} \Delta \theta_j^\rho \\ \Delta V_j^\rho / V_j^\rho \\ \Delta \phi_\rho \\ \Delta B_{eq}^\rho \end{bmatrix}^{(i)} \quad (2.15)$$

2.2.4 VSC with combined voltage and power flow control

The combined control of voltage and power flow uses the Jacobian in eq.(2.15) and changes the regulated voltage to the corresponding amplitude modulation ratio in the same way as discussed in section 2.2.2.

2.3 Test cases

Based on an existing three phase power flow computer program written in MATLAB, the code has been extended to solve the test cases discussed below. All test cases described in this chapter are small networks and serve the purpose to test the program for accuracy of results. A larger test network will be solved once accuracy has been verified. These differences arise during the iterative solution due to the fact that calculated powers are computed with voltages which are not quite correct. Upon convergence, the calculated powers match the regulated powers.

2.3.1 Power flow with cable parameters

The first test case is a three node network where the transmission lines are modeled with cable parameters extracted from a cable catalog. The cable models will be explained in Chapter 4.

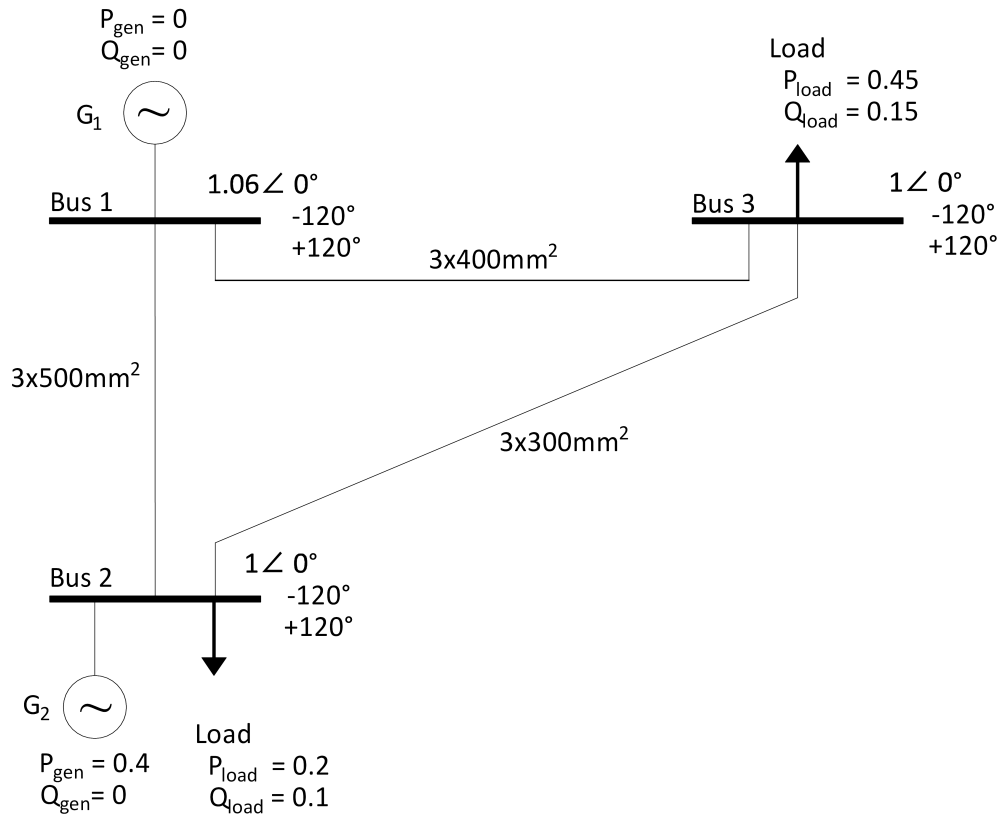


Figure 2.4 Configuration of the network for Test Case 2.1

Bus 1 contains the slack generator. Bus 2 is a PV-bus, it contains a synchronous generator which delivers 0.4 p.u. active power to the network and a balanced three-phase load of 0.2 p.u. of MW and 0.1 p.u. of MVar. A balanced voltage magnitude is kept a 1 p.u. Bus 3 is a PQ-bus, it contains a balanced load of 0.45 p.u. of MW and 0.15 p.u. of MVar. All transmission lines are modeled as three single conductor cables, placed in buried ducts and with a flat configuration, as shown in Fig.(2.5). The serie impedance and shunt admittance are given below in per-unit values.

$$Z_{3 \times 500} = \begin{bmatrix} 0.0094 + 0.0149j & 0.0034 - 0.0012j & 0.0026 - 0.0004j \\ 0.0034 - 0.0012j & 0.0088 + 0.0142j & 0.0034 - 0.0012j \\ 0.0026 - 0.0004j & 0.0034 - 0.0012j & 0.0093 + 0.0148j \end{bmatrix}$$

$$Y_{3 \times 500} = \begin{bmatrix} 0.0745j & -0.0015j & -0.0007j \\ -0.0015j & 0.0745j & -0.0015j \\ -0.0007j & -0.0015j & 0.0745j \end{bmatrix}$$

$$Z_{3 \times 400} = \begin{bmatrix} 0.0102 + 0.0154j & 0.0034 - 0.0012j & 0.0026 - 0.0004j \\ 0.0034 - 0.0012j & 0.0097 + 0.0147j & 0.0034 - 0.0011j \\ 0.0026 - 0.0004j & 0.0034 - 0.0011j & 0.0102 + 0.0153j \end{bmatrix}$$

$$Y_{3 \times 400} = \begin{bmatrix} 0.0232j & -0.0001j & -0.0001j \\ -0.0001j & 0.0232j & -0.0001j \\ -0.0001j & -0.0001j & 0.0232j \end{bmatrix}$$

$$Z_{3 \times 300} = \begin{bmatrix} 0.0114 + 0.0158j & 0.0034 - 0.0011j & 0.0026 - 0.0004j \\ 0.0034 - 0.0011j & 0.0108 + 0.0152j & 0.0034 - 0.0011j \\ 0.0026 - 0.0004j & 0.0034 - 0.0011j & 0.0113 + 0.0158j \end{bmatrix}$$

$$Y_{3 \times 300} = \begin{bmatrix} 0.0640j & -0.0011j & -0.0005j \\ -0.0011j & 0.0640j & -0.0011j \\ -0.0005j & -0.0011j & 0.0640j \end{bmatrix}$$

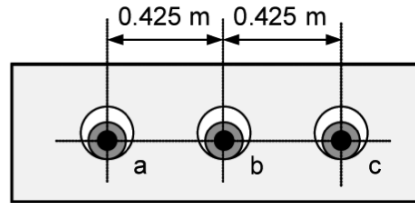


Figure 2.5 Test Case 2.1: Cable installation configuration in underground ducts

The solution is reached in 2 iterations, with the following results:

Table 2.2 Test Case 2.1: Voltage Magnitudes

VM (p.u.)	Phase a	Phase b	Phase c
Bus 1	1.0600	1.0600	1.0600
Bus 2	1.0000	1.0000	1.0000
Bus 3	1.0256	1.0240	1.0241

Table 2.3 Test Case 2.1: Voltage Angles

VA (°)	Phase a	Phase b	Phase c
Bus 1	0.0000	-120.0000	120.0000
Bus 2	1.5233	-118.8187	121.3488
Bus 3	0.4369	-119.6278	120.5464

Table 2.4 Test Case 2.1: Forward Power Flows

PQ (p.u.)	Phase a	Phase b	Phase c
Bus 1-2	0.2428+3.9147j	0.3508+3.6264j	0.3300+3.5100j
Bus 1-3	0.5047+2.0463j	0.5150+2.0698j	0.4136+1.9866j
Bus 2-3	0.2939-1.8155j	0.1857-1.5254j	0.1139-1.4950j

Table 2.5 Test Case 2.1: Reversed Power Flows

PQ (p.u.)	Phase a	Phase b	Phase c
Bus 2-1	-0.1305-3.7757j	-0.2595-3.5055j	-0.2316-3.3955j
Bus 3-1	-0.4732-2.0086j	-0.4844-2.0277j	-0.3811-1.9480j
Bus 3-2	-0.2672+1.8002j	-0.1686+1.4971j	-0.0951+1.4653j

Table 2.6 Test Case 2.1: Power losses

PQlosses (p.u.)	Phase a	Phase b	Phase c
Bus 1-2	0.1123+0.1390j	0.0913+0.1209j	0.0984+0.1145j
Bus 1-3	0.0315+0.0377j	0.0306+0.0421j	0.0325+0.0386j
Bus 2-3	0.0266-0.0153j	0.0171-0.0283j	0.0188-0.0296j

In this test case we can see that the state variables, the voltage magnitudes and the voltage angles, are rather good balanced. The unbalance is more clearly when we look at the power flows. The reactive power flow immediately stands out: the reactive power flow is up to ten times bigger than the active power flow! This effect

origins in the capacitive charged underground transmission lines and is strengthened by the relatively low load. When a higher inductive load is connected this effect will reduce.

2.3.2 Power flows with VSC

The test system in this section is aimed at verifying the accuracy of the three phase power flow computer program with VSC modeling facilities. The results are checked using an equivalent positive-sequence power flow computer program. For this test the zero sequence impedance of the transmission line is made equal to the positive sequence impedance. Three different situations are described:

- a) No control
- b) Voltage control at the sending end
- c) Voltage control at the sending end and power flow control at the receiving end.

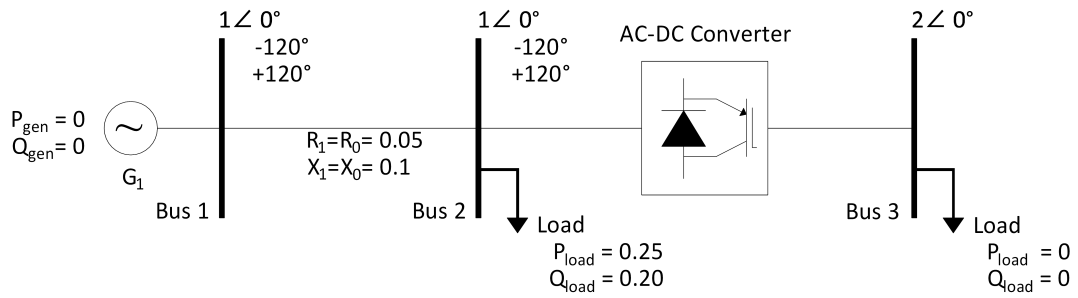


Figure 2.6 Configuration of the network for Test Case 2.2

a) No control

When there is no VSC control the Newton-Raphson solution converges in 5 iterations, yielding following results:

Table 2.7 Test Case 2.2a: Voltage Magnitudes

VM (p.u.)	Phase a	Phase b	Phase c
Bus 1	1.0000	1.0000	1.0000
Bus 2	1.0946	1.0946	1.0946
Bus 3	2.0000	2.0000	2.0000

Table 2.8 Test Case 2.2a: Voltage Angles

VA (°)	Phase a	Phase b	Phase c
Bus 1	0.0000	-120.0000	120.0000
Bus 2	-4.8276	-124.8276	115.1724
Bus 3	-5.6819	-5.6819	-5.6819

The amplitude modulation ratio m_ρ is 1 in all three phases and the phase shifting angle ϕ_ρ is 0. Also, the equivalent susceptance B_{eq}^ρ is 0.5 p.u. in every phase, which is the initial declared value. These variables are not actually included in the calculation because there is no VSC control.

The power flows and power losses in the transmission line and VSC are given in tables (2.9)-(2.11):

Table 2.9 Test Case 2.2a: Forward Power Flows

PQ (p.u.)	Phase a	Phase b	Phase c
Bus 1-2	0.3740-1.0946j	0.3740-1.0946j	0.3740-1.0946j
Bus 2-3	0.0571-1.4284j	0.0571-1.4284j	0.0571-1.4284j

Table 2.10 Test Case 2.2a: Reversed Power Flows

PQ (p.u.)	Phase a	Phase b	Phase c
Bus 2-1	-0.3071+1.2284j	-0.3071+1.2284j	-0.3071+1.2284j
Bus 3-2	0.0000+0.8489j	0.0000+0.8489j	0.0000+0.8489j

Table 2.11 Test Case 2.2a: Power losses

PQlosses (p.u.)	Phase a	Phase b	Phase c
Bus 1-2	0.0669+0.1338j	0.0669+0.1338j	0.0669+0.1338j
Bus 2-3	0.0571-0.5795j	0.0571-0.5795j	0.0571-0.5795j

b) Voltage control at the VSC's sending end

Voltage control is enabled to regulate voltage at the sending end of the VSC. This changes the use of state variable from V_2^ρ to m_ρ . The Newton-Raphson method now converges in 6 iterations, with the voltage solution given in tables (2.12)-(2.14).

Table 2.12 Test Case 2.2b: Voltage Magnitudes

VM (p.u.)	Phase a	Phase b	Phase c
Bus 1	1.0000	1.0000	1.0000
Bus 2	1.0500	1.0500	1.0500
Bus 3	2.0000	2.0000	2.0000

Table 2.13 Test Case 2.2b: Voltage Angles

VA (°)	Phase a	Phase b	Phase c
Bus 1	0.0000	-120.0000	120.0000
Bus 2	-3.5161	-123.5161	116.4839
Bus 3	-4.1730	-4.1730	-4.1730

Table 2.14 Test Case 2.2b: Amplitude modulation ratio

m_ρ	Phase a	Phase b	Phase c
VSC 1	0.9265	0.9265	0.9265

The voltage magnitude at the regulated bus is set to 1.05 p.u. The complex amplitude modulation ratio is $0.9265\angle 0^\circ$ in phase *a*. In phase *b* and *c* the amplitude modulation ratio is shifted over -120° and 120° , respectively.

The power flows and power losses of the transmission line and VSC are given in tables (2.15)-(2.17).

Table 2.15 Test Case 2.2b: Forward Power Flows

PQ (p.u.)	Phase a	Phase b	Phase c
Bus 1-2	0.3231-0.6418j	0.3231-0.6418j	0.3231-0.6418j
Bus 2-3	0.0473-0.8934j	0.0473-0.8934j	0.0473-0.8934j

Table 2.16 Test Case 2.2b: Reversed Power Flows

PQ (p.u.)	Phase a	Phase b	Phase c
Bus 2-1	-0.2973-0.6934j	-0.2973-0.6934j	-0.2973-0.6934j
Bus 3-2	0.0000+0.3222j	0.0000+0.3222j	0.0000+0.3222j

Table 2.17 Test Case 2.2b: Power losses

PQlosses (p.u.)	Phase a	Phase b	Phase c
Bus 1-2	0.0258+0.0516j	0.0258+0.0516j	0.0258+0.0516j
Bus 2-3	0.0473-0.5712j	0.0473-0.5712j	0.0473-0.5712j

c) Combined voltage control at the sending end and power flow control at the receiving end

The power flow control introduces 2 additional state variables per phase: ϕ_ρ and B_{eq}^ρ . The solution is reached in 6 iterations. The voltage solution is given in tables (4.15)-(2.19) whereas the VSC parameters are given in tables (2.20)-(2.22). It should be noticed that the voltages are balanced and regulated. The complex amplitude modulation ratios consists out of an amplitude modulation ratio, which has the same value compared to the case with only voltage regulation, and a phase shifting angle which is introduced by regulation the power flow at the receiving end. The voltage angles at bus 3 remain zero to ensure no reactive power flow. The lag of bus 3 compared to bus 2 is denoted by the phase shifting angle of the VSC.

Table 2.18 Test Case 2.2c: Voltage Magnitudes

VM (p.u.)	Phase a	Phase b	Phase c
Bus 1	1.0000	1.0000	1.0000
Bus 2	1.0500	1.0500	1.0500
Bus 3	2.0000	2.0000	2.0000

Table 2.19 Test Case 2.2c: Voltage Angles

VA (°)	Phase a	Phase b	Phase c
Bus 1	0.0000	-120.0000	120.0000
Bus 2	-3.5161	-123.5161	116.4839
Bus 3	0.0000	0.0000	0.0000

Table 2.20 Test Case 2.2c: Amplitude modulation ratio

m_ρ	Phase a	Phase b	Phase c
VSC 1	0.9265	0.9265	0.9265

Table 2.21 Test Case 2.2c: Phase shifting angle

ϕ (°)	Phase a	Phase b	Phase c
VSC1	-4.1731	-124.1731	115.8269

Table 2.22 Test Case 2.2c: Equivalent susceptance

B_{eq} (p.u.)	Phase a	Phase b	Phase c
VSC1	0.7502	0.7502	0.7502

Tables (2.23)-(2.25) give the power flows and power losses. The power regulation is obviously noticeable in the power flows: in the forward power flows we see a reactive power flow from the VSC to bus 2. In the reversed power flow we notice there is no active or reactive power flowing from bus 3 back to bus 2.

Table 2.23 Test Case 2.2c: Forward Power Flows

PQ (p.u.)	Phase a	Phase b	Phase c
Bus 1-2	0.3231-0.6418j	0.3231-0.6418j	0.3231-0.6418j
Bus 2-3	0.0473-0.8934j	0.0473-0.8934j	0.0473-0.8934j

Table 2.24 Test Case 2.2c: Reversed Power Flows

PQ (p.u.)	Phase a	Phase b	Phase c
Bus 2-1	-0.2973+0.6934j	-0.2973+0.6934j	-0.2973+0.6934j
Bus 3-2	0.0000-0.0000j	0.0000-0.0000j	0.0000-0.0000j

Table 2.25 Test Case 2.2c: Power losses

PQlosses (p.u.)	Phase a	Phase b	Phase c
Bus 1-2	0.0258+0.0516j	0.0258+0.0516j	0.0258+0.0516j
Bus 2-3	0.0473-0.8934j	0.0473-0.8934j	0.0473-0.8934j

If we compare the 3 different scenarios, we notice that the power flow is the highest when no control option is enabled. Also the power losses are the highest in this scenario. When voltage control is enabled we see a reduction in the voltage angles on both sides of the VSC. That's the reason why the power flow and also the power losses decreases. If power flow control is enabled together with voltage control we notice that there is no reactive power at the DC side of the VSC. This is also visible in the voltage angle of bus 3. The reactive power is provided by the equivalent susceptance. The phase shifting angle now takes account for the lagging of bus 3 to bus 2.

3. STATE ESTIMATION

Selective monitoring of power systems provides the data needed for dispatch and control of the grid. More recently the interconnected networks have become more complex, and similarly their operation. To help avoid major issues and blackouts, an extensive supervisory control and data acquisition (SCADA) system has been enrolled. The SCADA system helps the computer-based operation in control centers. The data bank created with all the measurement data can be used by various application programs [9].

Before any security or control action can take place, a reliable estimate of the current state of the system must be determined. The input quantities for traditional power flow calculations are restricted to the active and reactive power for load buses, voltage magnitude and active power for generator buses. If one of the quantities is not available, the conventional power flow solution cannot be obtained [9]. Even worse, if those quantities are available but a gross error is represented in one of the measurements, the power flow results are completely useless. These limitations can be avoided by the use of state estimation.

State estimation provides a technique to represent the state of a power system by calculating the voltage magnitude and voltage angles in all nodal points of the network with the use of different measurements. In contrast to the power flow algorithm, there are no restrictions for the measurements as long as they can be linked to one or more state variables. In this master thesis we will assume following types of quantities are measured:

- voltage magnitudes, represented by V_x
- injected active and reactive powers, represented respectively by P_x and Q_x
- active and reactive power flow, represented by P_{xy} and Q_{xy}
- inverse active and reactive power flow, represented by P_{yx} and Q_{yx}

3.1 Three phase state estimation algorithm

The measurements are collected and provided to a state estimation program. Every measurement z can be seen as a result of a function H with one or more state variables x . It's a well-known fact that measurements never display the exact value, but instead they represent the result in a certain range of confidence. Taking into account the possible error of the measurement, each measured value can be seen as the real value with a certain, unknown, error e . Combining a set of m measurements in a system with n state variables, in matrix form, yields following formulation:

$$z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_m \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & \dots & h_{1n} \\ h_{21} & h_{22} & h_{23} & \dots & h_{2n} \\ h_{31} & h_{32} & h_{33} & \dots & h_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{m1} & h_{m2} & h_{m3} & \dots & h_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_m \end{bmatrix} \quad (3.1)$$

Or in shorter notation:

$$z = Hx + e \quad (3.2)$$

In order to fully represent the state of the grid, a redundant set of measurements is necessary. Also note that these expressions are nonlinear and the h -coefficients are determined by the network constraints.

Due to the errors in the measurement, the exact values of the state variables cannot be determined. Instead the state variables will be estimated by minimizing the errors. The errors can be seen as the difference between the measured values and the actual, unknown, values.

$$e = z - z_{true} = z - Hx \quad (3.3)$$

The estimated values of x are represented by \hat{x} . After estimating \hat{x} , the estimated value of \hat{e} can be computed.

$$\hat{e} = z - \hat{z}_{true} = z - H\hat{x} = e - H(\hat{x} - x) \quad (3.4)$$

3.1.1 Errors

If the estimated error can be minimized to zero, the estimated state variables represent the true values of the system. The error will be minimized by using the method of the least squares. This method prevents that positive and negative errors cancel each other out, which would be the case if we just make the sum of the errors. Since not every meter is build equally, some measurements will have greater accuracy then others. To treat measurements with greater accuracy more favorable then less accurate measurements, each error will be multiplied with a weighting factor w . This leads us to the objective function f which should be minimized: [11]

$$f = \sum_{j=1}^m w_j e_j^2 \quad (3.5)$$

We assume that the measurement errors are distributed according to a normal or Gaussian distribution. The probability density function (pdf) for a normal distribution is given as follows

$$p(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{z-\mu}{\sigma}\right)^2} \quad (3.6)$$

where z is a random variable, in this case the measurement. The expected value of z is given by μ and is defined by

$$\mu = E[z] = \int_{-\infty}^{+\infty} z \cdot p(z) dz \quad (3.7)$$

The area under the pdf curve gives the probability associated with the corresponding interval of the horizontal axis. The complete area under the curve $p(z)$ equals 1, because the value of z certainly lies between the two extreme values. The expected value μ is often called the mean, since the values of z are symmetrically clustered around μ [11]. The parameter σ is the standard deviation, σ^2 is called the variance. The variance of a random value z is defined as the expected value of the squared difference between the value z and his expected value μ .

$$\sigma^2 = E[(z - \mu)^2] \quad (3.8)$$

If $\mu = 0$ and $\sigma = 1$ the distribution of the variable z is called the standard normal distribution. For each normal distribution can be proved that:

- 68.2 % of the measurements lies within $\mu \pm \sigma$

- 95.4 % of the measurements lies within $\mu \pm 2\sigma$
- 99.6 % of the measurements lies within $\mu \pm 3\sigma$

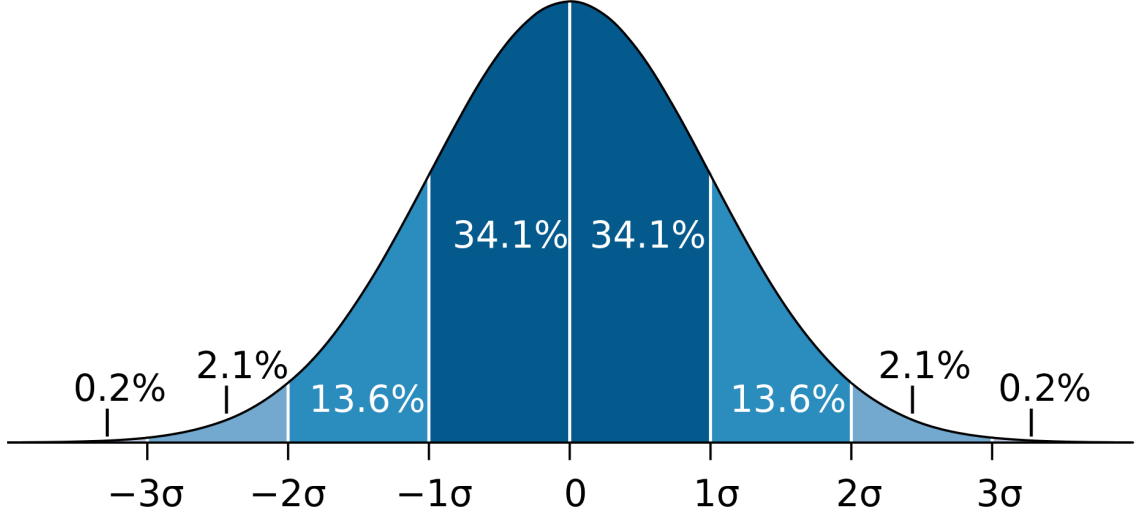


Figure 3.1 Standard Normal probability density function $p(z)$ [16]

Quite apart from the fact that the errors are distributed according to the standard normal distribution, the errors are independent of each other. This implies that each measurement has the same probability of taking a positive or a negative error and other measurements have no influence on the error. This can be used to define the weighting factor w . First we take the product of the vector e and its transposed e^T .

$$e \cdot e^T = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \end{bmatrix} \begin{bmatrix} e_1 & e_2 & \dots & e_m \end{bmatrix} = \begin{bmatrix} e_1^2 & e_1 e_2 & \dots & e_1 e_m \\ e_1 e_2 & e_2^2 & \dots & e_2 e_m \\ \vdots & \vdots & \ddots & \vdots \\ e_m e_1 & e_m e_2 & \dots & e_m^2 \end{bmatrix} \quad (3.9)$$

The expected value of the product of the vector e and its transposed can be found by calculating the expected value of each entry. Since the errors are independent, all non diagonal expected values are zero. All diagonal elements show the corresponding variances.

$$E[ee^T] = R = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n^2 \end{bmatrix} \quad (3.10)$$

The error of a meter with higher accuracy is spread in a smaller range and so the meter has a smaller variance. Since we want to treat a higher accuracy more favorable, the inverse of the variance can be used as a weighting factor and eq.(3.5) becomes

$$f = \sum_{j=1}^m \frac{e_j^2}{\sigma_j^2} \quad (3.11)$$

The necessary condition for minimizing f is:

$$\left. \frac{\partial f}{\partial x_j} \right|_{\hat{x}} = 2 \left(w_1 e_1 \frac{\partial e_1}{\partial x_j} + w_2 e_2 \frac{\partial e_2}{\partial x_j} + \cdots + w_m e_m \frac{\partial e_m}{\partial x_j} \right) \bigg|_{\hat{x}} = 0 \quad (3.12)$$

which can be written as

$$H_x^T W \hat{e} = 0 \quad (3.13)$$

where $W = R^{-1}$ is the diagonal matrix of weighting factors and H_x is the Jacobian. To solve this equation, the Newton-Raphson technique will be applied. The linearized equation can be found by substitution of eq.(3.4) into eq.(3.13). After elaboration, the following equation for the mismatch vector is achieved:

$$\Delta x^{(i)} = \underbrace{(H_x^{(i)T} R^{-1} H_x^{(i)})^{-1}}_{G^{(i)}} H_x^{(i)T} R^{-1} e = G^{(i)-1} H_x^{(i)T} R^{-1} e^{(i)} \quad (3.14)$$

where $\Delta x^{(i)}$ is the mismatch between the estimation of the state variables and the real value of the state variables at iteration i . Matrix G is called the gain matrix. Each iteration the elements of the Jacobian H_x are recalculated. The iterative process will continue until the mismatch is smaller than a specified precision ε for two successive iterations.

3.1.2 Jacobian matrix H_x

Unlike the square Jacobian J of the power flow equations, the Jacobian H_x in state estimation has always more rows than columns since there need to be more measurements than state variables to have redundancy. Each row of H_x corresponds with one of the measured quantities, where superscript ρ denotes the phase a, b or c :

- the voltage magnitude $|V_k^\rho|$ at a bus k
- the active power P_k^ρ injected at a bus k

- the reactive power Q_k^ρ injected at a bus k
- the active power flow P_{km}^ρ or P_{mk}^ρ between buses k and m
- the reactive power flow Q_{km}^ρ or Q_{mk}^ρ between buses k and m

If $|V_k^\rho|$, P_k^ρ and Q_k^ρ are measured at every bus in the network and all the power flows P_{km}^ρ , P_{mk}^ρ , Q_{km}^ρ and Q_{mk}^ρ are measured, then the matrix H_x is full. When one of the quantities is not measured, the row will be deleted and the matrix is no longer full. The ratio of the rows to the columns is called the redundancy factor. When the number of measurements equals the number of state variables, the redundancy is lost and H_x is square. When the redundancy is lost, every measurement is needed to determine the state variables. We need to avoid this in order to get control over possible bad data in the measurements.

Although we do not define different types of buses in state estimation, one bus needs to be defined as the slack bus. The voltage angle of the slack bus will be used as a reference angle and will be set to 0° , 120° or 240° for respectively phase a , b and c . The voltage magnitude does not need to be specified, this makes the slack bus different from the slack bus used in power flow. Since the voltage magnitude is not specified it will be used in the Jacobian matrix H_x .

For the ease of use the set of measurements will be grouped as follows: first all the voltage magnitudes, next the active and reactive injected powers and last the active and reactive power flows. Symbolically the Jacobian can be represented as follows:

$$Q_{km}^\rho = -(V_k^\rho)^2 \left(\frac{B_{km}^{\rho'}}{2} - B_{km}^\rho \right) + V_k^\rho V_m^\rho \{ G_{km}^\rho \sin(\theta_k^\rho - \theta_m^\rho) - B_{km}^\rho \cos(\theta_k^\rho - \theta_m^\rho) \} \quad (3.19)$$

Once the Jacobian is calculated, the state variables can be updated using eq.(3.14).

$$x^{(i+1)} = x^{(i)} + \Delta x^{(i)} = x^{(i)} + G^{(i)-1} H_x^{(i)T} R^{-1} e^{(i)} \quad (3.20)$$

3.1.3 Bad data

When the measured data is accurate, the state estimation yields good results. If one of the measurements is bad, the system should be able to detect the bad data and remove it from the calculations. The bad data detection is possible due to the redundancy of the measurements. If the redundancy is lost, bad data detection is not possible. Measurements are either critical or redundant. A measurement is critical if its elimination will result in an unobservable system. If the data is not critical, it will belong to the redundant data [11].

To identify the bad data, we rely on statistics. The sum of squares of independent random variables, distributed according to the normal distribution, has a χ^2 distribution with $m-n$ degrees of freedom. Where the former represents the number of measurements and the latter the number of state variables. To test if there is bad data included in the measurement data, the function f will be calculated. Then the value corresponding with probability p and $(m-n)$ degrees of freedom from the χ^2 distribution table will be presented. This is the value $\chi_{(m-n),\alpha}^2$, where α refers to the confidence interval. In this thesis we use a confidence interval of 0.95, which means we have a chance of 95 % that the real value is located in the area under the $\chi_{(m-n),\alpha}^2$ curve. The test for bad data is given by the following equation.

$$\hat{f} \geq \chi_{(m-n),\alpha}^2 \quad (3.21)$$

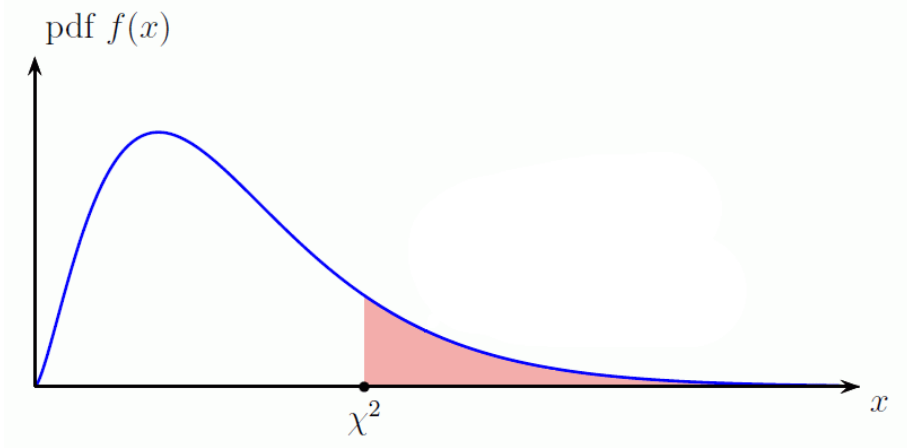


Figure 3.2 χ^2 -distribution for $m - n$ degrees of freedom and a confidence interval of α

If eq.(3.21) is satisfied then bad data is detected. The estimated value of f will be found in the colored area of figure 3.2. In case of bad data, the measurement with the largest standardized error will be removed from the measurement group. Since most meters measure not only one single phase but the three phases at once, the measurements of all the three phases will be removed in case of bad data detection in a single phase. After the corresponding measurement is deleted, all calculations will start over again with the new, smaller, set of measurement data. If the bad data test is not satisfied in the next run, the bad data is successfully identified and eliminated.

A flowchart for the state estimation algorithm with least squares is included on the next page.

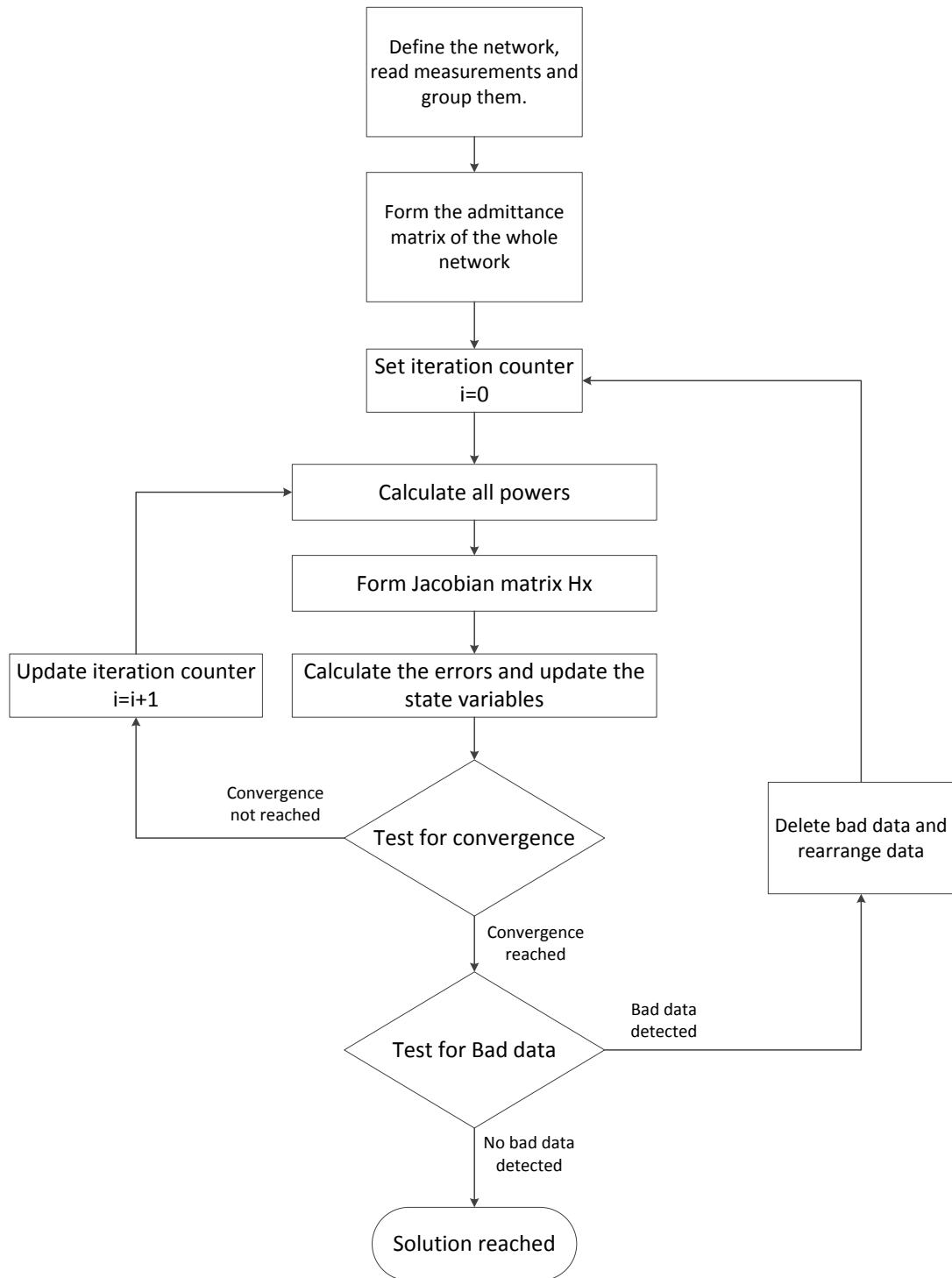


Figure 3.3 State Estimation algorithm flowchart

3.2 Three phase state estimation with VSC

When a VSC is included in the network it has no effect on the normal state estimation algorithm. However the VSC contribution needs to be added in the Jacobian matrix H_x . When a VSC is implemented in power flow studies, it results in extra state variables and extra equations. If we implement a VSC in state estimation it only results in extra state variables, since state estimation is based on measurements and the amount of measurements does not change. The number of state variables can rise depending on the control options.

3.2.1 VSC without control

When there is no control on the VSC, no extra state variables will be introduced. The degrees of freedom and the redundancy are not affected. The dimensions of the Jacobian remains the same. The VSC contributions will be added to the Jacobian in the same way as described in section 2.2.1.

3.2.2 VSC with voltage control

When voltage control is enabled on either the sending or the receiving end, the voltage magnitude of the controlled node is not a variable anymore. It will be replaced by the corresponding amplitude modulation ratio. The number of state variables stays the same. The degrees of freedom and the redundancy are not influenced.

3.2.3 VSC with power flow control

Two extra state variables per phase will be introduced when power flow control is enabled on the VSC. Six extra columns will be added to the Jacobian for each VSC, three for the phase shifting angle and three for the equivalent susceptances. Since the degrees of freedom is defined as the difference between the measurements and the state variables, a VSC with power flow control enabled will decrease the degrees of freedom. The redundancy factor will also decrease as an effect of the extra state variables.

The extended state variables vector is given by:

$$x = \begin{bmatrix} \Delta\theta_j^\rho \\ \Delta V_j^\rho / V_j^\rho \\ \Delta\phi_\rho \\ \Delta B_{eq}^\rho \end{bmatrix} \quad (3.22)$$

where of course the voltage angle of the slack bus is left out.

3.2.4 VSC with combined voltage and power flow control

In the case of a combined voltage and power flow control, the extended state variables vector is used. Once again the voltage magnitude of the voltage controlled node is replaced by the amplitude modulation ratio m_ρ . The combined control option has no extra effect on the degrees of freedom or the redundancy factor in comparison to power flow control.

3.3 Test cases

A three phase state estimation program was written using MATLAB. The following test networks were given to the program to verify the accuracy of the developed programs.

3.3.1 Checking the three phase state estimation program

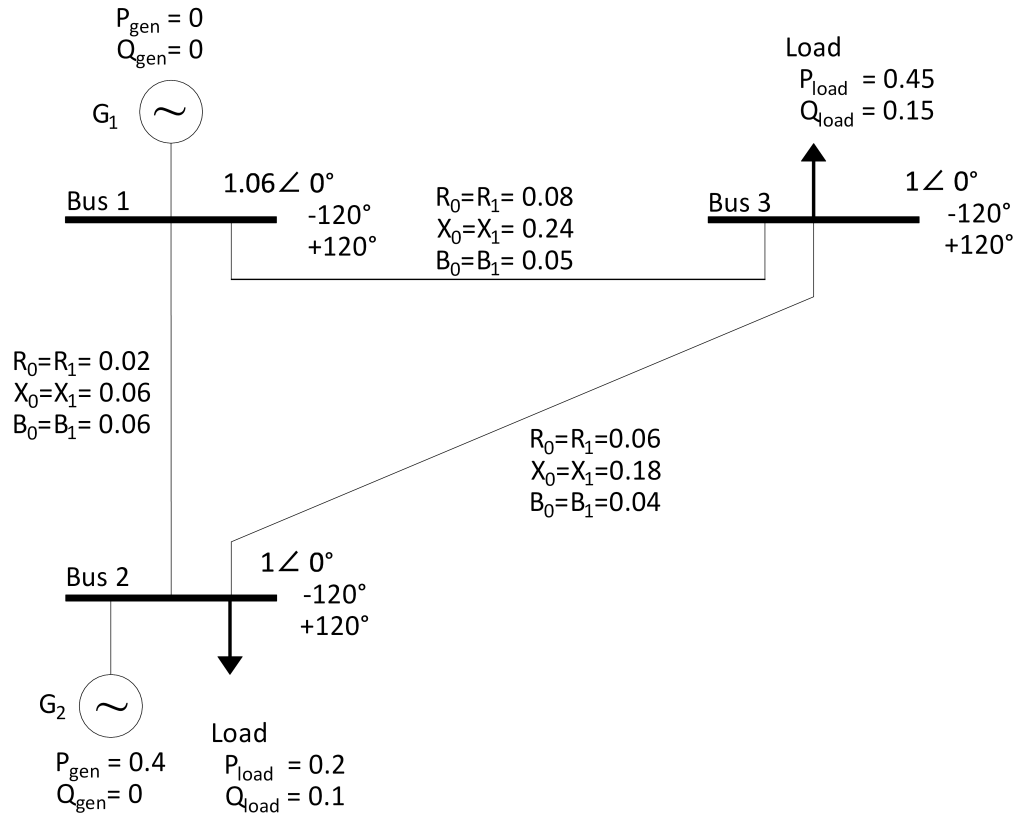


Figure 3.4 Configuration of the network for Test Case 3.1

In this test case we use a balanced 3-node network. Bus 1 is used as a slack bus. The serie impedance and shunt admittance of the transmission line is the same for the zero sequence as for the positive sequence in order to compare the results with a single phase state estimation program. The measurements used for this test case are the results from a three phase power program which was given the same test network. All results from the power flow program are used to have a full Jacobian matrix.

Table 3.1 Test Case 3.1: Measurement Data

Measurements		phase a	phase b	phase c
V_x	V_1	1.0600	1.0600	1.0600
	V_2	1.0000	1.0000	1.0000
	V_3	0.9995	0.9995	0.9995
P_x	P_1	0.2739	0.2739	0.2739
	P_2	0.2024	0.2024	0.2024
	P_3	-0.4470	-0.4470	-0.4470
Q_x	Q_1	1.1789	1.1789	1.1789
	Q_2	-1.1056	-1.1056	-1.1056
	Q_3	-0.1421	-0.1421	-0.1421
P_{xy}	P_{12}	0.0632	0.0632	0.0632
	P_{13}	0.2107	0.2107	0.2107
	P_{23}	0.2463	0.2463	0.2463
P_{yx}	P_{21}	-0.0439	-0.0439	-0.0439
	P_{31}	-0.2047	-0.2047	-0.2047
	P_{32}	-0.2423	-0.2423	-0.2423
Q_{xy}	Q_{12}	1.0075	1.0075	1.0075
	Q_{13}	0.1714	0.1714	0.1714
	Q_{23}	-0.0925	-0.0925	-0.0925
Q_{yx}	Q_{21}	-1.0131	-1.0131	-1.0131
	Q_{31}	-0.2065	-0.2065	-0.2065
	Q_{32}	0.0644	0.0644	0.0644

After providing the measurements to the three phase state estimator, following results were achieved in 5 iterations.

Table 3.2 Test Case 3.1: Estimated voltage magnitudes

VM (p.u.)	Phase a	Phase b	Phase c
Bus 1	1.0600	1.0600	1.0600
Bus 2	1.0000	1.0000	1.0000
Bus 3	0.9994	10.9994	0.9994

Table 3.3 *Test Case 3.1: Estimated voltage angles*

VA ($^{\circ}$)	Phase a	Phase b	Phase c
Bus 1	0.0000	-120.0000	120.0000
Bus 2	0.9206	-119.0794	120.9206
Bus 3	-1.8706	-121.8706	118.1294

To check the code this network was also provided to an existing single phase state estimator, which yielded following results after 6 iterations.

Table 3.4 *Test Case 3.1: Results in positive sequence*

	VM (p.u.)	VA ($^{\circ}$)
Bus 1	1.0600	0
Bus 2	1.0000	0.9205
Bus 3	0.9995	-1.8714

Looking at the voltage magnitudes we see perfect balanced results. Compared with the results from the single phase program we notice no difference. Generally speaking we can conclude that the three phase state estimation program works well and yields good results.

3.3.2 State estimation with VSC

As an extension to test case 2.2 in chapter 2 the three phase state estimation program is expanded to handle VSC's in the network. The program will be provided with the same network as in test case 2.2 and the output of the power flow program will be used as input for measurements. We will look at the same three control options.

a) No control

Without control, the solution is reached in 8 iterations and gave following results:

Table 3.5 Test Case 3.2a: Estimated voltage magnitudes

VM (p.u.)	Phase a	Phase b	Phase c
Bus 1	1.0000	1.0000	1.0000
Bus 2	1.0946	1.0946	1.0946
Bus 3	2.0000	2.0000	2.0000

Table 3.6 Test Case 3.2a: Estimated voltage angles

VA (°)	Phase a	Phase b	Phase c
Bus 1	0.0000	-120.0000	120.0000
Bus 2	-4.8276	-124.8276	115.1724
Bus 3	-5.6819	-5.6819	-5.6819

b) Voltage control on the sending end

When voltage control is enabled, it takes 5 iterations to converge.

Table 3.7 Test Case 3.2b: Estimated voltage magnitudes

VM (p.u.)	Phase a	Phase b	Phase c
Bus 1	1.0000	1.0000	1.0000
Bus 2	1.0500	1.0500	1.0500
Bus 3	2.0000	2.0000	2.0000

Table 3.8 Test Case 3.2b: Estimated voltage angles

VA (°)	Phase a	Phase b	Phase c
Bus 1	0.0000	-120.0000	120.0000
Bus 2	-3.5161	-123.5161	116.4839
Bus 3	-4.1730	-4.1730	-4.1730

Table 3.9 Test Case 3.2b: Estimated amplitude modulation ratio

m_ρ	Phase a	Phase b	Phase c
VSC 1	0.9265	0.9265	0.9265

c) Combined voltage control on the sending end and power flow control on the receiving end

Combined voltage and power flow control results in extra state variables, while the measurement inputs are not increased. The solution is reached in 17 iterations.

Table 3.10 *Test Case 3.2c: Estimated voltage magnitudes*

VM (p.u.)	Phase a	Phase b	Phase c
Bus 1	1.0000	1.0000	1.0000
Bus 2	1.0500	1.0500	1.0500
Bus 3	2.0000	2.0000	2.0000

Table 3.11 *Test Case 3.2c: Estimated voltage angles*

VA (°)	Phase a	Phase b	Phase c
Bus 1	0.0000	-120.0000	120.0000
Bus 2	-3.5161	-123.5161	116.4839
Bus 3	0.0000	0.0000	0.0000

Table 3.12 *Test Case 3.2c: Estimated amplitude modulation ratio*

m_ρ	Phase a	Phase b	Phase c
VSC 1	0.9265	0.9265	0.9265

Table 3.13 *Test Case 3.2c: Estimated phase shifting angle*

ϕ (°)	Phase a	Phase b	Phase c
VSC1	-4.1731	-124.1731	115.8269

Table 3.14 *Test Case 3.2c: Estimated equivalent susceptance*

B_{eq} (p.u.)	Phase a	Phase b	Phase c
VSC1	0.7502	0.7502	0.7502

When comparing these results with the results of test case 2.2, we see exactly the same values. This shows that both methods can be used to find the state variables. State estimation can provide the same results as power flow studies, while the measurement conditions are not as strict as in power flow studies.

4. MICRO-GRIDS

The definition of a micro-grid can be: a local group of electric power sources and loads that are connected to the normal grid and that function as expected. However in certain cases the micro-grid can be disconnected from the general grid and can operate autonomously. The micro grid can be used for dwellings, businesses and power resources. In the case of maintenance work, or worse, in the case of emergency's like a black out, the micro grid can be cut off from the grid. In this case the micro grid needs to provide all the power to maintain the demand of energy while operating in a stable way. Another aspect of a micro grid is that power plants are distributed all over the grid. Since most power resources in a micro grid are based on renewable energy, they heavily depend on the weather conditions. This is why micro grids need some kind of power storage solution [12, 13, 14].

Probably the most commonly used distributed energy resource are solar panels. These photo-voltaic (PV) generators are relatively easy to implement and don't require lots of planning and approval of the local government like wind turbines do. As we all know photo-voltaic cells generate a DC power, while the micro-grid in itself is an AC system. To transfer the generated DC power to the AC micro-grid the use of a DC-AC converter is needed. In order to represent the inverter in power flow studies and later on in the state estimation, a power flow model is necessary.

4.1 VSC model for power flows

A Voltage Source Converter (VSC) model will be used to link the PV generators to the micro-grid. A VSC enables the user to control either the voltage, power flow or both at a certain point in the network. In the case of an inverter the voltage on AC side and the power flow on DC side will be controlled. This model can be extended to incorporate the model of a PV generator.

4.1.1 Three phase power flow model

In order to successfully implement the VSC in the power flow and state estimation program, a three phase model of a VSC is made using the techniques learned in [4] for a STATCOM. For a better understanding, the principle scheme of a three phase inverter is given below.

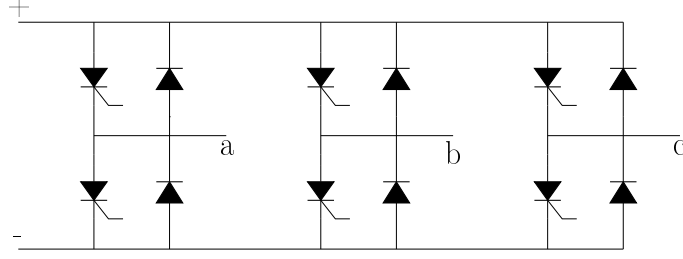


Figure 4.1 Three phase Inverter principle scheme

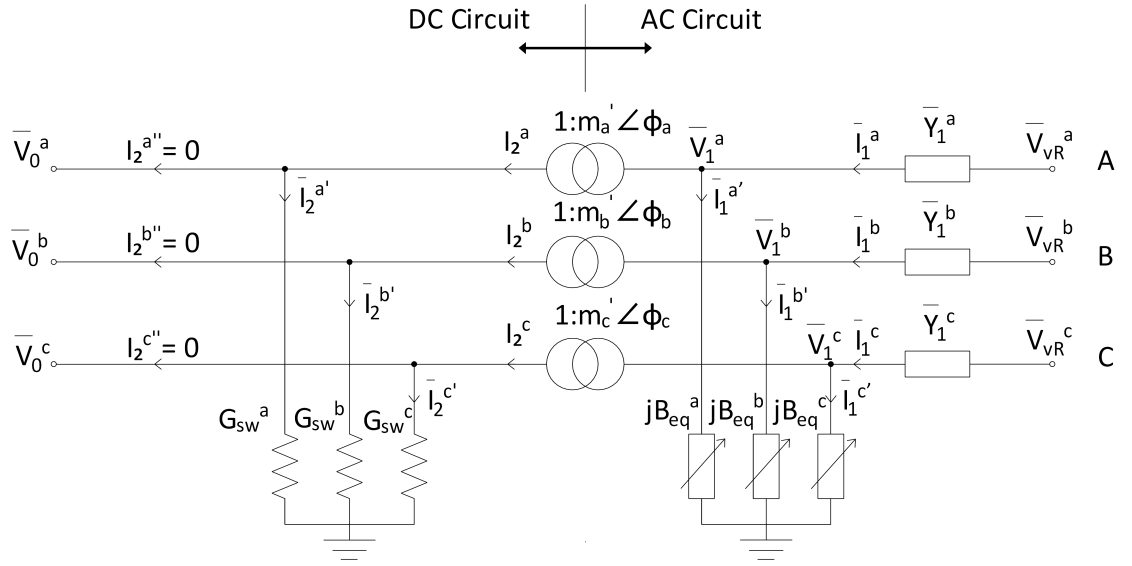


Figure 4.2 Three phase VSC power flow model

The primary and secondary side of each tap-changing transformer can be interpreted as the AC and the DC side of the inverter. The model also takes into account the phase shifting and the scaling factor ($1 : m'_\rho \angle \phi_\rho$). The conductance G_{sw}^ρ is current-dependent and is responsible for the switching losses in one leg of the inverter. In a first representation G_{sw}^ρ will be kept as a constant value. Under a constant

DC voltage and a constant load current G_{sw}^ρ yields a constant power loss [4]. On the DC-side of the transformer, no reactive power can be found. All the reactive power is generated in the AC-side. To take into account for the reactive power the susceptance B_{eq}^ρ is added. The susceptance is variable and takes whatever is necessary to meet Kirchoff's law. The equivalent admittance Y_1^ρ represents the internal ohmic and magnetic losses.

4.1.2 The nodal power equations

The basic relationships between the primary and secondary winding of each individual tap-changing transformer are given by the following equations:

$$\begin{aligned}\bar{V}_1^a &= m'_a e^{j\phi_a} \bar{V}_0^a \\ \bar{V}_1^b &= m'_b e^{j\phi_b} \bar{V}_0^b \\ \bar{V}_1^c &= m'_c e^{j\phi_c} \bar{V}_0^c\end{aligned}\tag{4.1}$$

The tap magnitude m_ρ corresponds to the amplitude modulation coefficient in phase ρ . In a three phase model $m'_\rho = k_1 m_\rho$, where $k_1 = \sqrt{\frac{3}{8}}$ and the linear range of modulation is $0 < m_\rho < 1$. The relation between the primary winding and the secondary winding can be noted as followed:

$$\begin{aligned}\frac{\bar{V}_1^a}{\bar{V}_0} &= k_1 m_a \angle \phi_a & \frac{\bar{V}_1^b}{\bar{V}_0} &= k_1 m_b \angle \phi_b & \frac{\bar{V}_1^c}{\bar{V}_0} &= k_1 m_c \angle \phi_c \\ \frac{\bar{I}_2^a}{\bar{I}_1^a - \bar{I}_1^{a'}} &= k_1 m_a \angle -\phi_a & \frac{\bar{I}_2^b}{\bar{I}_1^b - \bar{I}_1^{b'}} &= k_1 m_b \angle -\phi_b & \frac{\bar{I}_2^c}{\bar{I}_1^c - \bar{I}_1^{c'}} &= k_1 m_c \angle -\phi_c\end{aligned}\tag{4.2}\tag{4.3}\tag{4.4}$$

Notice that I_2 splits into I_2' and I_2'' , where the latter is zero during steady state. We assume no current is drawn by the DC voltage source.

Calculating the fictive injected current \bar{I}_0^ρ , knowing $I_2^{\rho''} = 0$ gives following equations:

$$\bar{I}_0^\rho = \bar{I}_2^{\rho'} - I_2^\rho = G_{sw}^\rho V_0^\rho - k_1 m_\rho \angle -\phi_\rho (\bar{Y}_1^\rho \bar{V}_{vR}^\rho - \bar{V}_1^\rho) - j B_{eq}^\rho V_1^\rho\tag{4.5}$$

Writing the equation as a function of V_{vR}^ρ and V_0^ρ .

$$\bar{I}_0^\rho = V_0^\rho G_{sw}^\rho + k_1 m_\rho (Y_1^\rho + B_{eq}^\rho) - k_1 m_\rho \angle -\phi_a \bar{Y}_1^\rho \bar{V}_{vR}^\rho\tag{4.6}$$

Writing down the current equations at AC-side:

$$\bar{I}_1^\rho = \bar{Y}_1^\rho (\bar{V}_{vR}^\rho - \bar{V}_1^\rho) = \bar{Y}_1^\rho \bar{V}_{vR}^\rho - k_1 m_\rho \angle \phi_\rho \bar{Y}_1^\rho \bar{V}_0^\rho \quad (4.7)$$

and combining eq.(4.6) and eq.(4.7) to a matrix form.

$$\begin{bmatrix} \bar{I}_{vR}^a \\ \bar{I}_{vR}^b \\ \bar{I}_{vR}^c \\ \bar{I}_0^a \\ \bar{I}_0^b \\ \bar{I}_0^c \end{bmatrix} = \begin{bmatrix} \bar{Y}_1^a & 0 & 0 & -k_1 m_a \angle \phi_a \bar{Y}_1^a \\ 0 & \bar{Y}_1^b & 0 & -k_1 m_b \angle \phi_b \bar{Y}_1^b \\ 0 & 0 & \bar{Y}_1^c & -k_1 m_c \angle \phi_c \bar{Y}_1^c \\ -k_1 m_a \angle - \phi_a \bar{Y}_1^a & 0 & 0 & G_{sw}^a + k_1 m_a^2 (Y_1^a + j B_{eq}^a) \\ 0 & -k_1 m_b \angle - \phi_b \bar{Y}_1^b & 0 & G_{sw}^b + k_1 m_b^2 (Y_1^b + j B_{eq}^b) \\ 0 & 0 & -k_1 m_c \angle - \phi_c \bar{Y}_1^c & G_{sw}^c + k_1 m_c^2 (Y_1^c + j B_{eq}^c) \end{bmatrix} \begin{bmatrix} \bar{V}_{vR}^a \\ \bar{V}_{vR}^b \\ \bar{V}_{vR}^c \\ \bar{V}_0^a \\ \bar{V}_0^b \\ \bar{V}_0^c \end{bmatrix} \quad (4.8)$$

The nodal power equations can be calculated as followed:

$$\begin{bmatrix} \bar{S}_{vR}^\rho \\ \bar{S}_0^\rho \end{bmatrix} = \begin{bmatrix} \bar{V}_{vR}^\rho & \\ 0 & \bar{V}_0^\rho \end{bmatrix} \begin{bmatrix} \bar{I}_{vR}^{\rho*} \\ \bar{I}_0^{\rho*} \end{bmatrix} \quad (4.9)$$

After some algebra, following equations appear:

$$\begin{aligned} P_{vR}^\rho &= G_1^\rho V_{vR}^{\rho^2} - k_1 m_\rho V_0^\rho V_{vR}^\rho [G_1^\rho \cos(\theta_{vR}^\rho - \theta_0^\rho - \phi_\rho) + B_1^\rho \sin(\theta_{vR}^\rho - \theta_0^\rho - \phi_\rho)] \\ Q_{vR}^\rho &= -B_1^\rho V_{vR}^{\rho^2} - k_1 m_\rho V_0^\rho V_{vR}^\rho [G_1^\rho \sin(\theta_{vR}^\rho - \theta_0^\rho - \phi_\rho) - B_1^\rho \cos(\theta_{vR}^\rho - \theta_0^\rho - \phi_\rho)] \end{aligned} \quad (4.10)$$

$$\begin{aligned} P_0^\rho &= V_0^{\rho^2} [G_{sw}^\rho + (k_1 m_\rho)^2 G_1^\rho] - k_1 m_\rho' V_0^\rho V_{vR}^\rho [G_1^\rho \cos(\theta_0^\rho - \theta_{vR}^\rho - \phi_\rho) + B_1^\rho \sin(\theta_0^\rho - \theta_{vR}^\rho - \phi_\rho)] \\ Q_0^\rho &= -(k_1 m_\rho)^2 V_0^{\rho^2} (B_1^\rho + B_{eq}^\rho) - k_1 m_\rho V_0^\rho V_{vR}^\rho [G_1^\rho \sin(\theta_0^\rho - \theta_{vR}^\rho - \phi_\rho) - B_1^\rho \cos(\theta_0^\rho - \theta_{vR}^\rho - \phi_\rho)] \end{aligned} \quad (4.11)$$

With the formulas given in eq.(4.10) and eq.(4.11) the power flows from the DC-side to the AC-side, and vice-versa, can be calculated. The solution will be found using the Newton-Raphson method, which means we have to form a Jacobian matrix once again. The state variables in this equivalent circuit are: θ_{vR}^ρ , V_{vR}^ρ , m_ρ , ϕ_ρ and B_{eq}^ρ where ρ represents the three phases a , b and c . The angle $\theta_0^\rho = 0$ and is used as a reference for the phase angles. The set of linearized equations regarding the power flow model, with voltage control and power flow control enabled, are presented in

matrix form below.

$$\begin{bmatrix} \Delta P_{vR}^\rho \\ \Delta Q_{vR}^\rho \\ \Delta P_0^\rho \\ \Delta Q_0^\rho \\ \Delta P_{0-vR}^\rho \\ \Delta Q_{0-vR}^\rho \end{bmatrix} = \begin{bmatrix} \frac{\partial P_{vR}^\rho}{\partial \theta_{vR}^\rho} & \frac{\partial P_{vR}^\rho}{\partial m_\rho} m_\rho & \frac{\partial P_{vR}^\rho}{\partial \theta_0} & \frac{\partial P_{vR}^\rho}{\partial V_0} V_0 & \frac{\partial P_{vR}^\rho}{\partial \phi_a} & 0 \\ \frac{\partial Q_{vR}^\rho}{\partial \theta_{vR}^\rho} & \frac{\partial Q_{vR}^\rho}{\partial m_\rho} m_\rho & \frac{\partial Q_{vR}^\rho}{\partial \theta_0} & \frac{\partial Q_{vR}^\rho}{\partial V_0} V_0 & \frac{\partial Q_{vR}^\rho}{\partial \phi_a} & 0 \\ \frac{\partial P_0^\rho}{\partial \theta_{vR}^\rho} & \frac{\partial P_0^\rho}{\partial m_\rho} m_\rho & \frac{\partial P_0^\rho}{\partial \theta_0} & \frac{\partial P_0^\rho}{\partial V_0} V_0 & \frac{\partial P_0^\rho}{\partial \phi_a} & 0 \\ \frac{\partial Q_0^\rho}{\partial \theta_{vR}^\rho} & \frac{\partial Q_0^\rho}{\partial m_\rho} m_\rho & \frac{\partial Q_0^\rho}{\partial \theta_0} & \frac{\partial Q_0^\rho}{\partial V_0} V_0 & \frac{\partial Q_0^\rho}{\partial \phi_a} & \frac{\partial Q_0^\rho}{\partial B_{eq}^\rho} \\ \frac{\partial P_{0-vR}^\rho}{\partial \theta_{vR}^\rho} & \frac{\partial P_{0-vR}^\rho}{\partial m_\rho} m_\rho & \frac{\partial P_{0-vR}^\rho}{\partial \theta_0} & \frac{\partial P_{0-vR}^\rho}{\partial V_0} V_0 & \frac{\partial P_{0-vR}^\rho}{\partial \phi_a} & 0 \\ \frac{\partial Q_{0-vR}^\rho}{\partial \theta_{vR}^\rho} & \frac{\partial Q_{0-vR}^\rho}{\partial m_\rho} m_\rho & \frac{\partial Q_{0-vR}^\rho}{\partial \theta_0} & \frac{\partial Q_{0-vR}^\rho}{\partial V_0} V_0 & \frac{\partial Q_{0-vR}^\rho}{\partial \phi_a} & \frac{\partial Q_{0-vR}^\rho}{\partial B_{eq}^\rho} \end{bmatrix} \cdot \begin{bmatrix} \Delta \theta_{vR}^\rho \\ \Delta m_\rho / m_\rho \\ \Delta \theta_0 \\ \Delta V_0 / V_0 \\ \Delta \phi_\rho \\ \Delta B_{eq}^\rho \end{bmatrix} \quad (4.12)$$

The voltage control is enabled on the sending end of the VSC. On the receiving end the power flow control is active. Depending on the control options the Jacobian matrix can be smaller than showed above. When power flow control is not enabled, ϕ_ρ and B_{eq}^ρ will disappear from the state variables vector. This will reduce the Jacobian with two rows and columns since the mismatch power equations are not needed.

4.1.3 The mismatch power flow equations

As seen in eq.(4.12) two mismatch power flow equations were added to the matrix form: ΔP_{0-vR}^ρ and ΔQ_{0-vR}^ρ . The active, respectively reactive power flow mismatches are only necessary if power flow is controlled on either the AC side or the DC side. Similar to the other power mismatches the power flow mismatches are defined as the mismatch of the regulated power flow and the calculated power flow.

$$\begin{aligned} \Delta P_{0-vR}^\rho &= P_{0-vR,reg}^\rho - P_{0-vR,cal}^\rho \\ \Delta Q_{0-vR}^\rho &= Q_{0-vR,reg}^\rho - Q_{0-vR,cal}^\rho \end{aligned} \quad (4.13)$$

Depending on which end the power flow will be controlled the sign of ΔP_{0-vR}^ρ will be different. In practice only power flow control on the DC-side can be achieved. If the power flow would be controlled on the AC side, the power losses are not known. The program divergences instead of reaching the solution.

With the use of Kirchhoff's current law the power flow mismatches can be found using figure (4.3). We assume there is no load on the DC side, only reactive power on the AC side and power flow control on the DC side. Writing down the fictitious power

injections at the DC side gives following results:

$$\begin{aligned}\Delta P_{0-vR}^p &= P_{0,load}^p - P_0^p = P_{vR}^p \\ \Delta Q_{0-vR}^p &= Q_{0,load}^p + Q_0^p = Q_0^p\end{aligned}\tag{4.14}$$

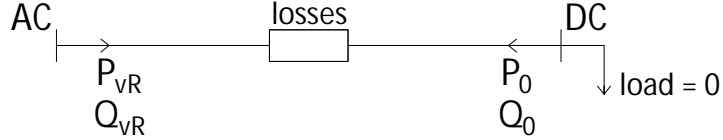


Figure 4.3 Graphical representation of the power flow mismatch function

Since there is no load, only power generation, on the DC side of the VSC, the power flow mismatches can be easily described as power injections.

4.2 Power flow model for a DC-DC Converter

The previous section describes a VSC model used to incorporate an AC-DC converter into a network. With some adjustments this model can also be used to describe a DC-DC converter. The DC-DC converter of choice is a buck-boost converter. This type of converter is a combination of a step-up and step-down converter which allows us to either lower or boost the DC voltage. Fig.(4.4) gives a principle scheme of the buck-boost converter.

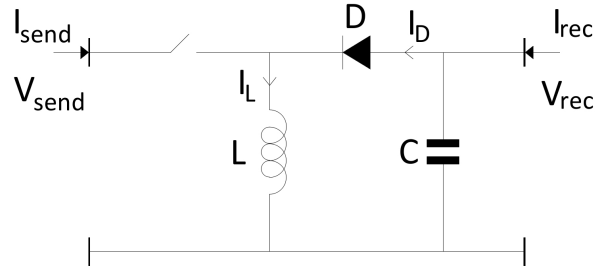


Figure 4.4 Principle scheme of the buck-boost Converter

The principle scheme allows us to obtain the basic relationship between the sending end and the receiving end rather easily:

$$V_{rec} = V_{send} \times \frac{\delta}{1 - \delta} \quad (4.15)$$

This relationship is valid if the converter operates in continuous mode (the current through the inductor never falls back to zero). The duty cycle δ gives the percentage of the period where the sending end gives energy to the inductor.

Knowing the basic equation of the VSC model:

$$V_{rec} = V_{send} \times k_1 m_a e^{j\phi} \quad (4.16)$$

we can adapt eq.(4.16) to describe a DC/DC converter. All we need to do is force the phase shifting angle ϕ to zero and change the two variables k_1 and m_a .

$$\begin{aligned} \phi &= 0 \\ k_1 &= 1 \\ m_a &= \frac{\delta}{1 - \delta} \end{aligned} \quad (4.17)$$

Also the limits of m_a need to be adapted to match the limits of the duty cycle δ .

4.2.1 Test case

To verify this model we will extend test case 2.2.

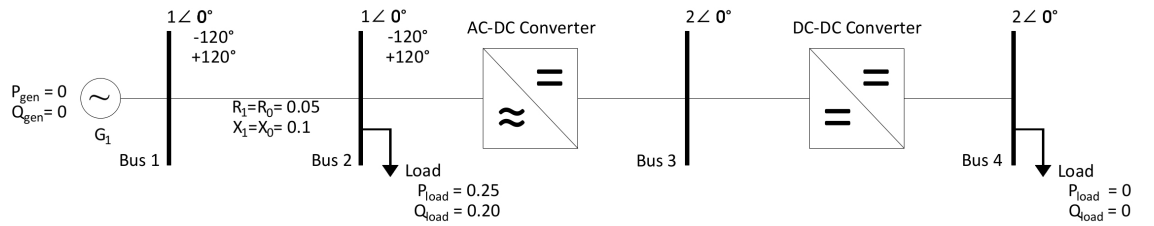


Figure 4.5 Configuration of the network for Test Case 4.1

Power flow solution

When there is no control on the converters, the Newton-Raphson solution converges in 9 iterations, yielding following results.

Table 4.1 Test Case 4.1a: Voltage Magnitudes

VM (p.u.)	Phase a	Phase b	Phase c
Bus 1	1.0000	1.0000	1.0000
Bus 2	1.0945	1.0945	1.0945
Bus 3	2.0000	2.0000	2.0000
Bus 4	2.0000	2.0000	2.0000

Table 4.2 Test Case 4.1a: Voltage Angles

VA (°)	Phase a	Phase b	Phase c
Bus 1	0.0000	-120.0000	120.0000
Bus 2	-4.8526	-124.8526	115.1474
Bus 3	-5.7248	-5.7248	-5.7248
Bus 4	-5.7306	-5.7306	-5.7306

Table 4.3 Test Case 4.1a: Amplitude modulation ratio

m_ρ	Phase a	Phase b	Phase c
VSC 1	1	1	1
VSC 2	1	1	1

Table 4.4 Test Case 4.1a: Phase shifting angle

ϕ (°)	Phase a	Phase b	Phase c
VSC1	0	-120.0000	120.0000
VSC2	0	0	0

As we can see the results in tables (4.1)-(4.4) are very similar to the results in test case 2.2a. The amplitude modulation ratio of VSC 2 needs to be converted to the duty cycle δ in order to have a useful meaning.

$$\delta = \frac{1}{1 + m_a} = \frac{1}{1 + 1} = 0.5 \quad (4.18)$$

The duty cycle δ is equal to 0.5 as initial declared. This value didn't change since there is no control.

The power flows and power losses in the transmission line and VSC are given in tables (4.5)-(4.7).

Table 4.5 Test Case 4.1a: Forward Power Flows

PQ (p.u.)	Phase a	Phase b	Phase c
Bus 1-2	0.3782-1.0954j	0.3782-1.0954j	0.3782-1.0954j
Bus 2-3	0.0611-1.4296j	0.0611-1.4296j	0.0611-1.4296j
Bus 3-4	0.0040-0.0004j	0.0040-0.0004j	0.0040-0.0004j

Table 4.6 Test Case 4.1a: Reversed Power Flows

PQ (p.u.)	Phase a	Phase b	Phase c
Bus 2-1	-0.3111+1.2296j	-0.3111+1.2296j	-0.3111+1.2296j
Bus 3-2	-0.0040+0.8505j	-0.0040+0.8505j	-0.0040+0.8505j
Bus 4-3	0.0000-1.9996j	0.0000-1.9996j	0.0000-1.9996j

Table 4.7 Test Case 4.1a: Power losses

PQlosses (p.u.)	Phase a	Phase b	Phase c
Bus 1-2	0.0671+0.1343j	0.0671+0.1343j	0.0671+0.1343j
Bus 2-3	0.0571-0.5791j	0.0571-0.5791j	0.0571-0.5791j
Bus 3-4	0.0040-2.0000j	0.0040-2.0000j	0.0040-2.0000j

Knowing the results of test case 2.2a, these results are almost identical, which is no surprise. We expect the same results, the extension with a DC-DC converter should only impact the power flows.

State estimation solution

The three phase state estimator will also be adapted to handle the two types of VSC's. Following results are achieved after 15 iterations:

Table 4.8 Test Case 4.1b: Estimated voltage magnitudes

VM (p.u.)	Phase a	Phase b	Phase c
Bus 1	1.0000	1.0000	1.0000
Bus 2	1.0945	1.0945	1.0945
Bus 3	2.0000	2.0000	2.0000
Bus 4	2.0000	2.0000	2.0000

Table 4.9 *Test Case 4.1b: Estimated voltage angles*

VA (°)	Phase a	Phase b	Phase c
Bus 1	0.0000	-120.0000	120.0000
Bus 2	-4.8526	-124.8526	115.1474
Bus 3	-5.7248	-5.7248	-5.7248
Bus 4	-5.7306	-5.7306	-5.7306

Table 4.10 *Test Case 4.1b: Estimated amplitude modulation ratio*

m_ρ	Phase a	Phase b	Phase c
VSC 1	1	1	1
VSC 2	1	1	1

Table 4.11 *Test Case 4.1b: Estimated phase shifting angle*

ϕ (°)	Phase a	Phase b	Phase c
VSC1	0	-120.0000	120.0000
VSC2	0	0	0

These results are identical to the results of the power flow study. We can say both programs are successfully adapted and ready to implement distributed PV generators. By representing a PV generator as a single phase generation, a first, rather simple, model can be achieved. This generator model is connected to a DC-DC converter to stabilize the voltage given to the inverter which injects the power into the AC network. This is done in test case 4.2.

4.3 Cable modeling

In order to represent transmission lines with the use of parameters found in catalogs, a cable model will be used. Since the amount of underground power cable installations is increasing due to environmental and technical reasons, underground power cables will be modeled as part of the micro-grid. Modeling underground cables is not that much different than modeling overhead lines. The formulas we will use are general textbook formulas which have proven their accuracy. Both the series impedance and the shunt admittance matrices are calculated in Ω/km .

4.3.1 Serie impedance matrix of power cables

First of all we want to calculate the impedance of the cable. To calculate the impedance matrix Caron's equations will be used: [7]

$$Z_{ii} = (r_i + k_1 f) + j k_2 f \ln \left(\frac{k_3}{\alpha_i R_i} \sqrt{\frac{\rho}{f}} \right) \quad (4.19)$$

$$Z_{ij} = k_1 f + j k_2 f \ln \left(\frac{k_3}{d_{ij}} \sqrt{\frac{\rho}{f}} \right) \quad (4.20)$$

where

- Z_{ii} = self impedance [Ω/km]
- Z_{ij} = mutual impedance [Ω/km]
- r_i = resistance of the conductor in [Ω/km]
- ρ = resistivity of the earth: $100 \Omega m$
- R_i = radius of the conductor in m
- d_{ij} = distance between conductors in m
- f = fundamental frequency: $50 Hz$
- $\alpha_i = e^{-\frac{1}{4}}$, conversion to Geometric Mean Radius (GMR)
- $k_1 = \pi^2 \times 10^{-4}$
- $k_2 = 4\pi \times 10^{-4} H/km$, magnetic permeability
- $k_3 = 658,87165$

Knowing these formulas, the primitive impedance matrix can be calculated.

$$Z_{primitive} = \begin{bmatrix} Z_{conductor-conductor}^{abc} & Z_{conductor-concentricwires}^{abc} \\ Z_{conductor-concentricwires}^{abc} & Z_{concentricwires-concentricwires}^{abc} \end{bmatrix} \quad (4.21)$$

When the concentric wires or the earth shield has been transposed and the main conductor is not transposed, the primitive matrix can be adapted to represent the

transposition. The self-resistances corresponding to concentric wires need to be adapted. All the self-resistances are now equal for all the three phases and are obtained as the average $= (r_{wa} + r_{wb} + r_{wc})/3$.

As a second option, the main conductor can be counter transposed to the concentric wires. All the elements of the primitive matrix are averaged, including the self resistance and reactance for conductors and concentric wires [6].

After adapting the primitive matrix, if necessary, Kron's reduction can be applied to achieve a 3×3 -matrix per transmission line. If needed, symmetrical components can be applied afterwards. Notice that if the main conductor is counter transposed, the decoupling in symmetrical components is perfect, but the losses are up to 20 % higher compared to transposed concentric wires and non-transposed conductors [6].

4.3.2 Shunt admittance matrix of power cables

As you might have noticed, until now there is no influence of the line conductance and susceptance. To involve the conductance and the susceptance, the admittance matrix will be calculated with the use of the method of potential coefficients. P_0 , P_1 and P_2 are potential matrices [7]. The subscripts 0, 1 and 2 are not related to zero, positive and negative sequence but are just a way of numbering the potential coefficients. The matrices look as followed:

$$P_0 = \begin{bmatrix} \ln(\frac{2h_1}{r_2}) & \ln(\frac{D_{12}}{d_{12}}) & \ln(\frac{D_{13}}{d_{13}}) \\ \ln(\frac{D_{21}}{d_{21}}) & \ln(\frac{2h_2}{r_2}) & \ln(\frac{D_{23}}{d_{23}}) \\ \ln(\frac{D_{31}}{d_{31}}) & \ln(\frac{D_{32}}{d_{32}}) & \ln(\frac{2h_3}{r_3}) \end{bmatrix} \quad (4.22)$$

$$P_1 = \begin{bmatrix} \ln(\frac{D_{ins,1}}{D_{shield,1}}) & 0 & 0 \\ 0 & \ln(\frac{D_{ins,2}}{D_{shield,2}}) & 0 \\ 0 & 0 & \ln(\frac{D_{ins,3}}{D_{shield,3}}) \end{bmatrix} \quad (4.23)$$

$$P_2 = \begin{bmatrix} \ln(\frac{D_{jack,1}}{D_{wires,1}}) & 0 & 0 \\ 0 & \ln(\frac{D_{jack,2}}{D_{wires,2}}) & 0 \\ 0 & 0 & \ln(\frac{D_{jack,3}}{D_{wires,3}}) \end{bmatrix} \quad (4.24)$$

where

- h_i = depth of the conductor under ground level

- r_i = radius of the conductor
- $D_{ij} = \sqrt{(y_i + y_j)^2 + (x_i - x_j)^2}$
- $d_{ij} = \sqrt{(y_i - y_j)^2 + (x_i - x_j)^2}$
- (x_i, y_i) = coordinates of the conductor
- $D_{ins,i}$ = insulation diameter of the cable
- $D_{shield,i}$ = shield diameter of the cable
- $D_{jack,i}$ = jacket diameter of the cable
- $D_{wires,i}$ = wire diameter of the cable

The admittance matrix will now be assembled with the 3 potential matrices.

$$Y_{cable} = j1000\omega_0 2\pi e_0 \begin{bmatrix} \frac{P_0}{\varepsilon_{r,ground}} + \frac{P_1}{\varepsilon_{r,XLPE}} + \frac{P_2}{\varepsilon_{r,Poly}} & \frac{P_2}{\varepsilon_{r,Poly}} \\ \frac{P_2}{\varepsilon_{r,Poly}} & \frac{P_2}{\varepsilon_{r,Poly}} \end{bmatrix}^{-1} \quad (4.25)$$

4.4 Micro-grid: test case

The final test case is a micro-grid represented by fig.(4.6). The network consists of a 132 kV loop. Node 1 acts as a slack bus and has a per unit value of 1.05 in each phase with a voltage angle of respectively 0°, −120° and 120°. The voltage magnitudes of all other AC buses is set to 1.0 p.u. and 2.0 p.u. for all DC buses. An opening point is available between nodes 11 and 14, but is not used in this test case. All transformers have grounded star connections on both the primary and secondary windings. The per phase transformer parameters are given in table 4.12.

Table 4.12 Test Case 4.2: Transformer Parameters

Transformer Parameters					
Transformer	Connecting nodes		Resistance (p.u.)	Reactance (p.u.)	Tap Position
1	N1	N2	0.012	0.120	0.97
2	N3	N4	0.030	0.300	0.99
3	N5	N6	0.015	0.150	0.95
4	N5	N7	0.030	0.300	1.05
5	N8	N9	0.015	0.150	0.95
6	N8	N10	0.030	0.300	1.05
7	N11	N12	0.030	0.300	0.99
8	N11	N13	0.030	0.300	1.03
9	N14	N15	0.030	0.300	0.93
10	N14	N16	0.030	0.300	1.02

The transmission lines are non-bonded single conductor cables, placed in buried ducts with a flat configuration as shown in Fig.(2.5). Using the cable parameters in table (4.13) the transmission line impedance and admittance matrix become:

$$Z_{abc} = \begin{bmatrix} 0.1700 + 0.1653j & 0.0945 - 0.0127j & 0.0732 - 0.0286j \\ 0.0945 - 0.0127j & 0.1583 + 0.1505j & 0.0945 - 0.0127j \\ 0.0732 - 0.0286j & 0.0945 - 0.0127j & 0.1700 + 0.1653j \end{bmatrix} \Omega/km$$

$$Y_{abc} = \begin{bmatrix} 71.99j & -3.22j & -1.51j \\ -3.22j & 72.10j & -3.22j \\ -1.51j & -3.22j & 71.99j \end{bmatrix} \mu S/km$$

Table 4.13 Test Case 4.2: Cable Parameters

Cable Parameters		
Element	Thickness (mm)	Diameter (mm)
Conductor, Copper, 5 segments	-	48.0
Conductor shield	3.2	54.4
Insulation,XLPE	15.8	86.0
Insulation screen	1.9	89.74
Concentric wires, Copper	2.0	93.8
Jacket, Polyethylene	4.6	103.0

The PV park injects 50 MW of active power and 25 MW of reactive power into the network. The wind turbine injects a total of 250 MW active power. The EV charging station is charging the batteries and draws a total of 50 MW. The STATCOMs are providing reactive power and are not controlled. The distribution systems and the housing estate are non-symmetrical loads; the total active and reactive powers are given in the following table:

Table 4.14 Test Case 4.2: Sources and Loads

Sources & Loads	Three-Phase Powers							
	Active Power (MW)				Active Power (MVar)			
	<i>a</i>	<i>b</i>	<i>c</i>	3ϕ	<i>a</i>	<i>b</i>	<i>c</i>	3ϕ
EV Charging Station	16.6667	16.6667	16.6667	50	0	0	0	0
Distribution System 1	84.4067	76.7333	69.0600	230.2	25.2633	22.9667	20.6700	68.9
Distribution System 2	69.0600	76.7333	84.4067	230.2	20.6700	22.9667	25.2633	68.9
Wind Turbine	83.3333	83.3333	83.3333	250	0	0	0	0
Housing Estate	31.6667	35.0000	33.3333	100	15.8333	17.5000	16.6667	50
PV Park	16.6667	16.6667	16.6667	50	8.3333	8.3333	8.3333	25

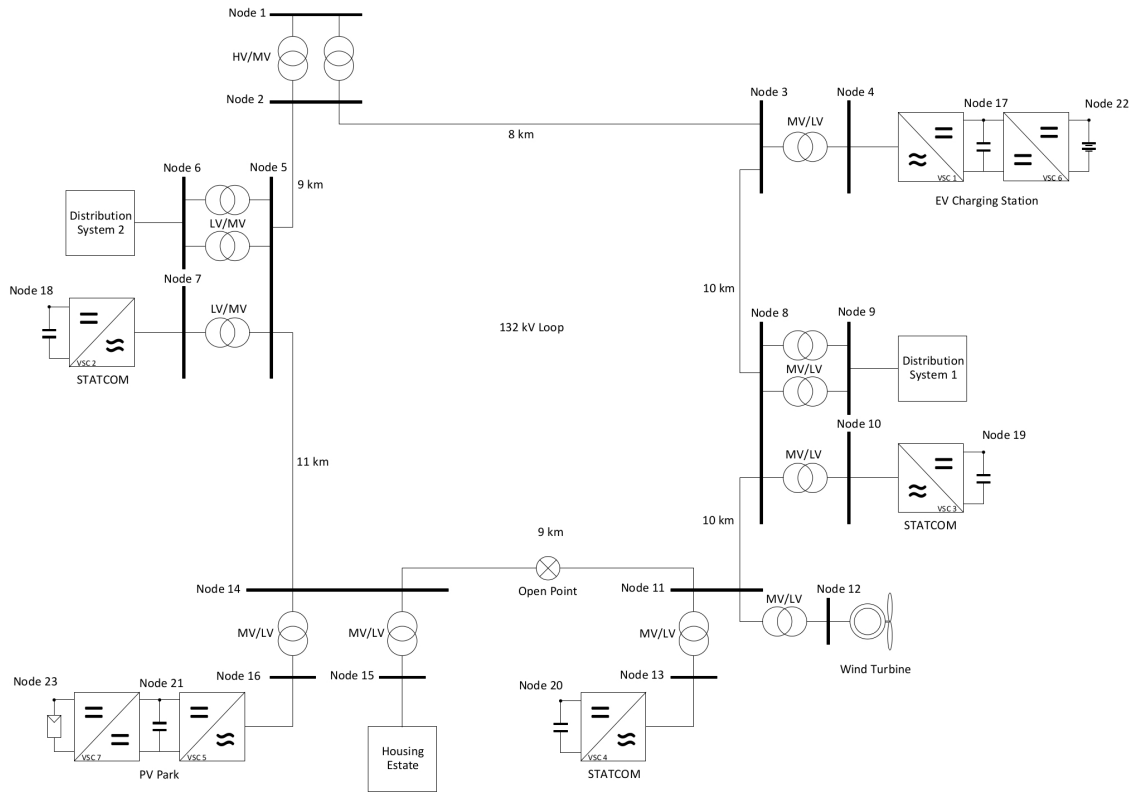


Figure 4.6 Configuration of the network for Test Case 4.2

4.4.1 Power flow solution

When providing the starting conditions of the test network to the Power Flow program, the EV Charging Station and the PV Park are modeled as a serie connection of an AC-DC converter and a DC-DC converter. On the receiving end of the DC-DC converter a equivalent single-phase load is connected in case of the Charging Station and an equivalent single-phase generator in case of the PV Park. We assume the PV generator provides a continuous power to the DC-DC converter and the inverter provides the required reactive power.

The solution is reached in 14 iterations, with the following results:

Table 4.15 *Test Case 4.2a: Voltage Magnitudes*

VM (p.u.)	Phase a	Phase b	Phase c
Node 1	1.0500	1.0500	1.0500
Node 2	1.1000	1.0971	1.0972
Node 3	1.0980	1.00974	1.1006
Node 4	1.1664	1.1661	1.1679
Node 5	1.0936	1.0926	1.0894
Node 6	1.1053	1.0982	1.0884
Node 7	1.0933	1.0923	1.0891
Node 8	1.0723	1.0746	1.0800
Node 9	1.0915	1.0929	1.0986
Node 10	1.0908	1.0985	1.1108
Node 11	1.0912	1.0926	1.0983
Node 12	1.1022	1.0995	1.1035
Node 13	1.1124	1.1095	1.1137
Node 14	1.1255	1.1227	1.1268
Node 15	1.0986	1.0963	1.0973
Node 16	1.1187	1.1085	1.1135
Node 17	1.1466	1.1452	1.1458
Node 18	2.0005	2.0004	2.0012
Node 19	1.8788	1.8771	1.8715
Node 20	1.8752	1.8775	1.8873
Node 21	1.9341	1.9293	1.9363
Node 22	2.0000	2.0000	2.0000
Node 23	2.0000	2.0000	2.0000

Table 4.16 Test Case 4.2a: Voltage Angles

VA (°)	Phase a	Phase b	Phase c
Node 1	0.0000	-120.0000	120.0000
Node 2	-7.2656	-127.4895	112.5715
Node 3	-7.9506	-128.1517	111.8248
Node 4	-10.7687	-130.9733	109.0228
Node 5	-8.1344	-128.2761	111.4854
Node 6	-13.1550	-133.8973	105.2256
Node 7	-8.5975	-128.7393	111.0222
Node 8	-8.3380	-128.5909	111.3568
Node 9	-14.5717	-134.2091	106.3837
Node 10	-8.8012	-129.0540	110.8937
Node 11	-7.7035	-127.9063	112.1103
Node 12	4.1815	-115.9593	123.9680
Node 13	-8.1858	-128.3885	111.6280
Node 14	-8.0551	-128.2163	111.6380
Node 15	-12.5842	-133.2800	106.8423
Node 16	-6.4030	-126.5639	113.2903
Node 17	-13.2690	-13.4756	-13.4682
Node 18	-9.0598	-9.2015	-9.4400
Node 19	-9.2634	-9.5163	-9.5686
Node 20	-8.6480	-8.8507	-8.8342
Node 21	-4.8853	-5.0486	-5.1933
Node 22	-13.9969	-14.2040	-14.1939
Node 23	-4.1885	-4.3533	-4.4973

Non of the VSCs in the network are controlled, this leaves all the amplitude modulation ratio's to 1 and the phase shifting angles the initial 0° , -120° , 120° for AC/DC converters. The duty cycle δ remains 0.5 for the DC-DC converters with no phase shifting angle. Tables (4.17)-(4.19) give the power flow and power losses in the transmission lines and in the VSCs.

Table 4.17 Test Case 4.2a: Forward Power Flows

PQ (p.u.)	Phase a	Phase b	Phase c
Node 2-3	0.5246-0.2398j	0.4954-0.2570j	0.4505-0.2787j
Node 2-5	0.6221-0.1007j	0.6860-0.0660j	0.7213-0.0431j
Node 3-8	0.3362+0.0173j	0.3071+0.0013j	0.2621-0.0259j
Node 5-14	-0.0934 - 0.1407j	-0.1074-0.1470j	-0.1540-0.1736j
Node 8-11	-0.5332-0.0958j	-0.4840-0.0675j	-0.4511-0.0512j
Node 11-14	0.2648-0.0112j	0.3150-0.0164j	0.3477+0.0352j
VSC1	0.1836-0.2420j	0.1836-0.2428j	0.1835-0.2380j
VSC2	0.0129-0.2095	0.0128-0.2092j	0.0128-0.2079j
VSC3	0.0128-0.2087j	0.0129-0.2093j	0.0130-0.2115j
VSC4	0.0136-0.2221j	0.0136-0.2210j	0.0137-0.2226j
VSC5	-0.1498-0.2633j	-0.1498-0.2666j	-0.1498-0.2651j
VSC6	0.1682-0.0132j	0.1682-0.0132j	0.1682-0.0076j
VSC7	-0.1651-0.0362j	-0.1651-0.0402j	-0.1651-0.0384j

Table 4.18 Test Case 4.2a: Reversed Power Flows

PQ (p.u.)	Phase a	Phase b	Phase c
Node 3-2	-0.5218+0.2039j	-0.4927+0.2207j	-0.4476+0.2436j
Node 5-2	-0.6181+0.0628j	-0.6826+0.0280j	-0.7148+0.0101j
Node 8-3	-0.3355-0.0666j	-0.3060-0.0513j	-0.2605-0.0242j
Node 14-5	0.0925+0.0843j	0.1077+0.0898j	0.1558+0.1183j
Node 11-8	0.5366+0.0504j	0.4864+0.0210j	0.4537+0.0049j
Node 14-11	-0.2649-0.0342j	-0.3143-0.0620j	-0.3453-0.791j
VSC1	-0.1682+0.0122j	-0.1682+0.0132j	-0.1682+0.0076j
VSC2	0.0000+0.0000j	0.0000+0.0000j	0.0000+0.0000j
VSC3	0.0000+0.0000j	0.0000+0.0000j	0.0000+0.0000j
VSC4	0.0000+0.0000j	0.0000+0.0000j	0.0000+0.0000j
VSC5	0.1651+0.0362j	0.1651+0.0402j	0.1651+0.0384j
VSC6	-0.1667-0.6524j	-0.1667-0.6514j	-0.1667-0.6570j
VSC7	0.1667-0.6283j	0.1667-0.6243j	0.1667-0.6261j

Table 4.19 Test Case 4.2a: Power losses

PQlosses (p.u.)	Phase a	Phase b	Phase c
Node 2-3	0.0028-0.0359j	0.0026-0.0364j	0.0029-0.0351j
Node 2-5	0.0039-0.0378j	0.0034-0.0380j	0.0065-0.0330j
Node 3-8	0.0007-0.0492j	0.0011-0.0500j	0.0015-0.0501j
Node 5-14	-0.0009-0.0564j	0.0002-0.0571j	0.0019-0.0553j
Node 8-11	0.0034-0.0453j	0.0024-0.0465j	0.0027-0.0463j
Node 11-14	-0.0001-0.0454j	0.0007-0.0457j	0.0024-0.0439j
VSC1	0.0154-0.2298j	0.0154-0.2297j	0.0153-0.2304j
VSC2	0.0129-0.2095j	0.0128-0.2092j	0.0128-0.2079j
VSC3	0.0128-0.2087j	0.0129-0.2093j	0.0130-0.2115j
VSC4	0.0136-0.2221j	0.0136-0.2210j	0.0137-0.2226j
VSC5	0.0153-0.2270j	0.0154-0.2264j	0.0153-0.2267j
VSC6	0.0015-0.6645j	0.0015-0.6645j	0.0015-0.6645j
VSC7	0.0016-0.6645j	0.0016-0.6645j	0.0016-0.6645j

Looking at the power flows we can clearly see that VSCs 2-4 acts as STATCOMs since they don't draw active power from the DC side to the AC side, only the reactive power needed for the distribution systems and the housing estate are provided. VSC 1 draws active power from the 132 kV loop to the batteries in order to charge them and the reactive power to compensate for the losses in the converters. The PV park injects balanced active power into the loop. The transmission line between nodes 8 and 11 transports the highest amount of power in this network and also has the highest losses. The least loaded section appears to be between nodes 5 and 14. The directions of the power flow can be found by looking at tables (4.17) and (4.18) or at figure (4.7). The power need for Distribution System 1 is coming from the wind turbine at node 11 and from the slack bus. Distribution System 2 is fed by the PV park and the slack bus. The slack bus also provides power to the EV charging station. The housing estate is fed by the PV park and the wind turbine.

Table 4.20 *Test Case 4.2b: Estimated voltage magnitudes*

VM (p.u.)	Phase a	Phase b	Phase c
Node 1	1.0499	1.0457	1.0576
Node 2	1.0980	1.0869	1.0943
Node 3	1.0951	1.0873	1.0952
Node 4	1.1680	1.1629	1.1654
Node 5	1.0892	1.0798	1.0837
Node 6	1.0970	1.0788	1.0750
Node 7	1.0849	1.0758	1.0745
Node 8	1.0869	1.0821	1.0905
Node 9	1.0843	1.0797	1.0928
Node 10	1.0869	1.0758	1.0791
Node 11	1.0996	1.0897	1.0987
Node 12	1.1130	1.1081	1.1217
Node 13	1.1286	1.1259	1.1371
Node 14	1.0945	1.0847	1.0915
Node 15	1.1121	1.0919	1.1006
Node 16	1.1433	1.1370	1.1387
Node 17	2.0073	2.0031	2.0010
Node 18	1.8627	1.8485	1.8418
Node 19	1.8691	1.8470	1.8491
Node 20	1.9430	1.9940	1.9637
Node 21	1.9880	1.9828	1.9803
Node 22	2.0079	2.0044	2.0007
Node 23	1.9963	1.9922	1.9885

Table 4.21 Test Case 4.2b: Estimated voltage angles

VA (°)	Phase a	Phase b	Phase c
Node 1	0.0000	-120.0000	120.0000
Node 2	-7.8712	-128.2322	112.1799
Node 3	-8.6935	-128.9892	111.3769
Node 4	-11.7576	-132.3128	107.9810
Node 5	-8.8034	-129.2025	111.0842
Node 6	-14.0577	-135.2617	104.6870
Node 7	-9.8507	-130.5558	110.5431
Node 8	-9.1934	-129.4744	110.9340
Node 9	-15.7747	-135.4127	105.6996
Node 10	-10.6493	-130.4787	110.0098
Node 11	-8.4110	-128.7196	111.6721
Node 12	3.9248	-116.2479	123.6365
Node 13	-8.0656	-128.4333	111.4425
Node 14	-8.7638	-129.1297	111.2124
Node 15	-13.4665	-134.5041	106.2696
Node 16	-7.4227	-127.8898	112.7109
Node 17	-14.3888	-15.0865	-14.8514
Node 18	-10.5882	-11.4386	-9.9637
Node 19	-11.6044	-11.1997	-10.6750
Node 20	-8.1318	-8.8297	-8.8950
Node 21	-6.0867	-6.6137	-5.8562
Node 22	-15.1382	-15.8607	-15.6375
Node 23	-5.4214	-5.9590	-5.1701

5. SUMMARY

The first objective of this thesis was to develop a three phase state estimation program. The second objective was to expand the three phase state estimation program to incorporate distributed PV generators.

After studying three phase power flows and single phase state estimation, a single phase state estimation program has successfully been expanded to a three phase state estimation program. The three phase state estimation program has been tested and its accuracy has been verified. To incorporate distributed PV generators, a three phase VSC has been derived from a single phase STATCOM model. This model is implemented in both the three phase power flow program and the three phase state estimation program. Once this VSC model has been tested to act as an AC-DC converter, a suggestion is made to use this VSC model to describe a DC/DC converter. Both programs have been adapted to use this VSC model both as an AC-DC and as a DC-DC converter. At this stage the programs are ready to incorporate distributed PV generators.

Several test cases have been assessed. Most of them are used to verify the program, but the last test case combines all previous test cases to provide a realistic scenario where both power flow and state estimation are used to calculate the state of the system. These two programs are also included in appendix B and C.

5.1 Recommendations for future work

In this thesis the PV generators are represented as a single phase generator, although a better model can be easily included. The most challenging part of including the converters has already been done and further expansion is rather straightforward at this point.

Since power flow and state estimation have a very wide scope, further research will

always be possible. Power flow and state estimation will keep updating as long as our grid will update and the state-of-the-art techniques will become more familiar. For example a coupling of different micro-grids with the use of a High Voltage DC-bus (HVDC) is already present in the single phase power flow studies [3]. This article can be found in appendix A. It would be really interesting to expand this multi-terminal HVDC system to a three phase representation.

BIBLIOGRAPHY

- [1] E. Acha, C. R. Fuerte-Esquivel, H. Ambriz-Perez, and C. Angeles-Camacho, *FACTS: Modelling and Simulation in Power Networks*. Wiley, 2004.
- [2] E. Acha, “Electric power systems: Power flow analysis,” Lecture notes Electric Power Systems (TUT), 2016.
- [3] E. Acha, L. M. Castro, and T. Rubbrecht, “Power flow solutions of ac/dc micro-grid structures,” in *Power Systems Computation Conference*, 2016.
- [4] E. Acha and B. Kazemtabrizi, “A new statcom model for power flows using the newton-raphson method,” *IEEE*, pp. 2455–2465, Aug. 2013.
- [5] G. Andersson, “Power system analysis,” Lecture notes Power Systems Analysis (ETH Zurich), 2012.
- [6] F. De León, M. L. Márquez-Asensio, and G. Álvarez Cordero, “Effect of conductor counter-transposition on the positive-sequence impedance and losses of cross-bonded cables,” *IEEE*, pp. 2060–2063, June 2011.
- [7] H. Dommel, “Overhead line parameters from handbook formulas and computer programs,” *IEEE*, pp. 366–372, Feb. 1985.
- [8] A. Gomez-Exposito, A. J. Conejo, and C. Canizares, *Electric Energy Systems: Analysis and Operation*. CRC Press, 2009.
- [9] J. J. Grainger and W. D. Stevenson Jr., *Power System Analysis*, international edition ed. McGraw-Hill, 1994.
- [10] G. W. Stagg and A. H. El-Abiad, *Computer Methods in Power System Analysis*, international student edition ed. McGraw-Hill, 1968.
- [11] G. Swiszcz, “On-line computer control of power systems,” Master’s thesis, University of Glasgow, Aug 2004.
- [12] “About microgrids.” [Online]. Available: <http://www.microgridinstitute.org/about-microgrids.html>
- [13] “About microgrids.” [Online]. Available: <https://building-microgrid.lbl.gov/about-microgrids>

- [14] “How microgrids work.” [Online]. Available: <http://www.energy.gov/articles/how-microgrids-work>
- [15] “Newton’s method.” [Online]. Available: <https://en.wikipedia.org/wiki/Newton's-method>
- [16] “Standard deviation.” [Online]. Available: <https://en.wikipedia.org/wiki/Standard-deviation>

Power Flow Solutions of AC/DC Micro-grid Structures

Enrique ACHA and Tom RUBBRECHT
Department of Electrical Engineering
Tampere University of Technology
Tampere, Finland

Luis M. CASTRO
Department of Electrical Engineering
National Autonomous University of Mexico
Mexico city, Mexico.

Abstract—This paper presents a new and general frame-of-reference for the unified, power flow solution of AC and DC micro-grids using the Newton-Raphson method, where the quadratic convergence towards the solution is preserved. The cornerstone of this modeling development in power flow theory is the so-called multi-terminal VSC-HVDC system. In this frame-of-reference, an AC micro-grid of arbitrary configuration is connected to the high-voltage side of the LTC transformer of a VSC station. In turn, the DC side of each VSC is linked to a DC system of arbitrary configuration. Any number of AC micro-grids may exist and the DC system may contain single load or generation points such as a PV installation. Each VSC model takes into account, in aggregated form, the phase-shifting and scaling nature of the PWM control. It also accounts for the VSC current design limits, PWM limits within the linear range, switching losses and ohmic losses.

Index Terms—Micro-grids, multi-terminal HVDC systems, Newton-Raphson method, power flows, VSC modeling

I. INTRODUCTION

The global electricity supply industry is undergoing unprecedented change to be able to cope with major challenges arising from an ageing infrastructure, market liberalisation and the availability of renewable generation. Over the past decade, the concepts of active networks, micro-grids and smart grids have been put forward as theoretical frameworks aimed at addressing these challenges [1]-[2]. It is said that a micro-grid will contain de-centralized electricity generation combined with on-site production of heat, bringing about substantial environmental benefits to society. With the use of modern technology, micro-grids enable the integration of renewable energy sources and achieve a good match between generation and load inside the micro-grid, reducing the impact on the neighboring electricity network.

In a more general sense, the consensus is that tomorrow's power grids must ensure secure and sustainable electricity supplies with low energy losses and low CO₂ emissions [3]. In Europe, these power grids should also comply with new policy imperatives, changing business frameworks and to incorporate the state-of-the-art information technology, communications technology and the latest generation of electrical equipment. Paramount in this array of new technologies is the ubiquitous power electronic converter, which power engineers have used in a variety of forms to

enable the instantaneous control of the voltage and current waveforms in the electrical power grid [4]. Power electronics converters are also used in grid connection of renewable sources of electrical energy and storage systems [5].

Current research efforts in the power electronics area concentrate on the development of modular, multi-level power electronic converters, SiC valves, self-monitoring and fault tolerant converters. It is argued that this will result in more efficient, scalable, reliable and inexpensive converters with longer lifetimes and improved performances, paving the way for the common place existence of AC/DC micro-grids. It is surmised that AC/DC micro-grids would be amenable to higher energy yields than AC micro-grids, reducing very considerably carbon footprints and using less material resources. It is envisaged that multi-terminal VSC-HVDC systems are very well placed to be the transmission structures that will be used in the next generation of micro-grids.

The design and operation of AC/DC micro-grids calls for the development of new models, methods and control techniques embedded in software. For instance, the operation of a multi-terminal VSC-HVDC-based micro-grid may be assessed by building a model that comprises a number of VSC units which is commensurate with the number of terminals in the HVDC system, suitably accommodated in an all-encompassing frame-of-reference. This paper introduces such a frame-of-reference, with particular reference to the power flow solution of micro-grids, using the Newton-Raphson algorithm which exhibits quadratic convergence owing to its true unified characteristics. The topic of multi-terminal VSC-HVDC power flows and transient simulations has received a fair amount of research attention over the past five years aimed at bulk power transmission [6-8], as opposed to micro-grid systems, which is the remit of this paper.

II. THE BASIC MODEL

The fundamental frequency, steady-state operation of a Multi-Terminal VSC-HVDC (MT-VSC-HVDC) system may be assessed by building a compound model that comprises a number of basic VSC models, which equals the number of terminals in the HVDC system. By way of example, the three-terminal VSC-HVDC system shown in Fig. 1 illustrates this concept where three AC micro-grids are connected asynchronously through a DC grid.

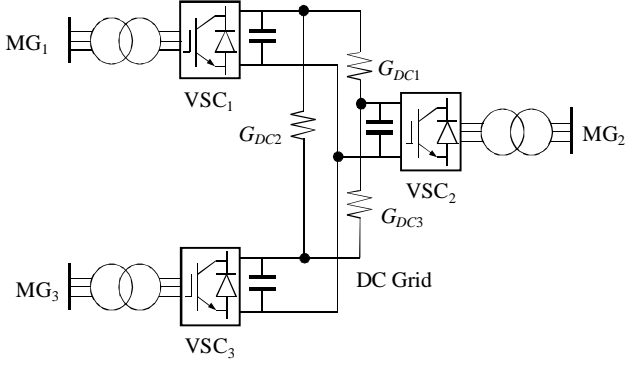


Figure 1. Three-terminal VSC-HVDC system

Each converter unit in the AC/DC system illustrated in Fig. 1 comprises a VSC, a phase reactor and a filter capacitor, in addition to the LTC transformer, to connect to the high-voltage AC network, as illustrated in Fig. 2.

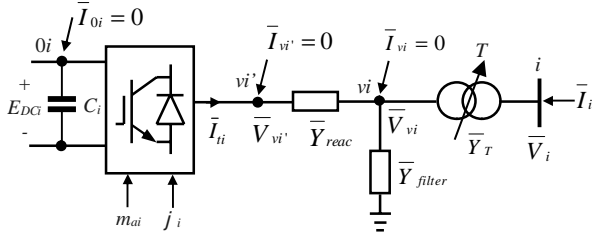


Figure 2. Generic VSC station i , including ancillary elements

The intended functionality of the phase reactor and filter capacitor is aimed at harmonic frequencies, to improve the quality of the voltage and current waveforms at the low-voltage side of the connecting transformer. However, their inductance and capacitance parameters affect also the fundamental frequency operation of the VSC station and require representation within the power flow formulation.

The model of the basic VSC unit is the kernel with which the three-terminal VSC-HVDC system shown in Fig. 1 is built. The kernel has been developed in [9] for the case of a STATCOM. It has the nodal admittance matrix given below:

$$\begin{bmatrix} \bar{Y}_{vi,vi} & \bar{Y}_{vi,vi'} & \bar{Y}_{vi,0i} \\ \bar{Y}_{vi,vi'} & \bar{Y}_{vi,vi'} & \bar{Y}_{vi,0i} \\ \bar{Y}_{vi,0i} & \bar{Y}_{vi,0i} & \bar{Y}_{vi,0i} \end{bmatrix} \begin{bmatrix} \bar{V}_{vi} \\ \bar{V}_{vi'} \\ \bar{V}_{0i} \end{bmatrix} = \begin{bmatrix} \bar{I}_{vi} \\ \bar{I}_{vi'} \\ \bar{I}_{0i} \end{bmatrix} \quad (1)$$

where $\bar{Y}_1 = 1/(R_1 + jX_1)$ and R_1 and X_1 account for the ohmic losses and the interface magnetics internal to the VSC. The

current-dependent resistor, G_{sw} , accounts for the converter switching power loss and B_{eq} is an equivalent susceptance which is responsible for the whole of the reactive power production in the VSC's valve set. The amplitude modulation index, m_{ai} , should be kept within its linear range ($0 < m_{ai} < 1$), for a smooth operation. The phase angle j_i is the phase angle of the complex voltage $\bar{V}_{vi'}$ relative to the system phase reference and $k_i = \sqrt{3}/8$ for cases of three-phase converters.

Owing to the nodal admittance nature of the basic VSC model, it becomes quite a straightforward matter to combine it with the representation of the smoothing line reactor and the shunt filter, given rise to a model where nodes vi , vi' and $0i$ are explicitly represented. However, since the external injected current at node vi' is nil then a more compact representation is arrived at by the mathematical elimination of node vi' , using Kron's reduction [10]:

$$\begin{bmatrix} \bar{Y}_{vi,vi} & \bar{Y}_{vi,0i} \\ \bar{Y}_{vi,0i} & \bar{Y}_{vi,0i} \end{bmatrix} \begin{bmatrix} \bar{V}_{vi} \\ \bar{V}_{0i} \end{bmatrix} = \begin{bmatrix} \bar{I}_{vi} \\ \bar{I}_{0i} \end{bmatrix} \quad (2)$$

where

$$\begin{aligned} \bar{Y}_{vi,vi} &= G_{vi,vi} + jB_{vi,vi} = \bar{Y}_{filter} + \bar{Y}_1 \bar{Y}_{reac} / (\bar{Y}_1 + \bar{Y}_{reac}) \\ \bar{Y}_{vi,0i} &= G_{vi,0i} + jB_{vi,0i} = -k_i m_{ai} \bar{Y}_1 \bar{Y}_{reac} / (\bar{Y}_1 + \bar{Y}_{reac}) \\ \bar{Y}_{0i,0i} &= G_{0i,0i} + jB_{0i,0i} = (G_{sw} + jk_i^2 m_{ai}^2 B_{eq}) + k_i^2 m_{ai}^2 \bar{Y}_1 \bar{Y}_{reac} / (\bar{Y}_1 + \bar{Y}_{reac}) \\ G_{sw} &= G_0 (|\bar{I}_{vi}| / I_{nom})^2 \end{aligned}$$

The conductance, G_0 , is derived at nominal DC voltage and rated current and remain constant.

Expression (2) is combined with the nodal transfer admittance matrix equation of the LTC transformer, which would be connected between nodes k and vi . The resulting nodal admittance matrix representing the VSC station with ancillary elements, shown in Fig. 2, is:

$$\begin{bmatrix} \bar{Y}_{vi,vi} & \bar{Y}_{vi,vi'} & \bar{Y}_{vi,0i} \\ \bar{Y}_{vi,vi'} & \bar{Y}_{vi,vi'} & \bar{Y}_{vi,0i} \\ \bar{Y}_{vi,0i} & \bar{Y}_{vi,0i} & \bar{Y}_{vi,0i} \end{bmatrix} \begin{bmatrix} \bar{V}_{vi} \\ \bar{V}_{vi'} \\ \bar{V}_{0i} \end{bmatrix} = \begin{bmatrix} \bar{I}_{vi} \\ \bar{I}_{vi'} \\ \bar{I}_{0i} \end{bmatrix} \quad (3)$$

where \bar{Y}_{ii} is the leakage admittance of the i -th LTC transformer and T_i is its tap.

III. THE THREE-TERMINAL POWER FLOW MODEL

By way of example, the nodal matrix equation for the three-terminal system of Fig. 1 is given in eqn. (4).

$$\begin{bmatrix} \bar{Y}_{v1,v1} & \bar{Y}_{v1,v1'} & \bar{Y}_{v1,01} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \bar{Y}_{v1,v1'} & \bar{Y}_{v1,v1'} & \bar{Y}_{v1,01} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \bar{Y}_{v1,01} & \bar{Y}_{v1,01} & \bar{Y}_{v1,01} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \bar{Y}_{v2,v2} & \bar{Y}_{v2,v2'} & \bar{Y}_{v2,02} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \bar{Y}_{v2,v2'} & \bar{Y}_{v2,v2'} & \bar{Y}_{v2,02} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \bar{Y}_{v2,02} & \bar{Y}_{v2,02} & \bar{Y}_{v2,02} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \bar{Y}_{v3,v3} & \bar{Y}_{v3,v3'} & \bar{Y}_{v3,03} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \bar{Y}_{v3,v3'} & \bar{Y}_{v3,v3'} & \bar{Y}_{v3,03} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \bar{Y}_{v3,03} & \bar{Y}_{v3,03} & \bar{Y}_{v3,03} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{Y}_{v4,v4} & \bar{Y}_{v4,v4'} & \bar{Y}_{v4,04} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{Y}_{v4,v4'} & \bar{Y}_{v4,v4'} & \bar{Y}_{v4,04} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{Y}_{v4,04} & \bar{Y}_{v4,04} & \bar{Y}_{v4,04} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{V}_{v1} \\ \bar{V}_{v1'} \\ \bar{V}_{01} \\ \bar{V}_{v2} \\ \bar{V}_{v2'} \\ \bar{V}_{02} \\ \bar{V}_{v3} \\ \bar{V}_{v3'} \\ \bar{V}_{03} \\ \bar{V}_{v4} \\ \bar{V}_{v4'} \\ \bar{V}_{04} \end{bmatrix} = \begin{bmatrix} \bar{I}_{v1} \\ \bar{I}_{v1'} \\ \bar{I}_{01} \\ \bar{I}_{v2} \\ \bar{I}_{v2'} \\ \bar{I}_{02} \\ \bar{I}_{v3} \\ \bar{I}_{v3'} \\ \bar{I}_{03} \\ \bar{I}_{v4} \\ \bar{I}_{v4'} \\ \bar{I}_{04} \end{bmatrix} \quad (4)$$

(a) Nodal power equations for the three-terminal system of Figure 1

The nodal power equations are derived by multiplying the nodal voltages by the conjugate of the nodal currents. Separation of the ensuing equations into real and imaginary parts yields the nodal active and reactive powers expressions.

The nodal power equations for one generic VSC station i – refer to Fig. 2 – are derived and then by suitable replacement of subscripts, the corresponding nodal power equations of the three VSC stations in Fig. 1 become readily available.

Hence, involving nodes i , vi and $0i$ in the circuit of Fig. 2 and the currents expressions in eqn. (3), we have,

$$\begin{aligned} \bar{S}_i &= \bar{V}_i \bar{I}_i^* = \bar{V}_i \left(\frac{E_{DC}}{Z_{DC}} + \frac{E_{DC}}{Z_{DC}} \right) \\ \bar{S}_{vi} &= \bar{V}_{vi} \bar{I}_{vi}^* = \bar{V}_{vi} \left(\frac{E_{DC}}{Z_{DC}} + \frac{E_{DC}}{Z_{DC}} \right) \\ \bar{S}_{0i} &= \bar{V}_{0i} \bar{I}_{0i}^* = \bar{V}_{0i} \left(\frac{E_{DC}}{Z_{DC}} + \frac{E_{DC}}{Z_{DC}} \right) \end{aligned} \quad (5)$$

Note that the complex nodal voltages may be expressed either in rectangular coordinates or in polar coordinates. In this paper the former representation will be used.

Following some complex number algebra, we have:

$$P_i = -k_i m_i E_{DC} \left(e_{vi} \cos j_i + B_{vi,j} \sin j_i \right) + G_{vi,j} (e_{vi}^2 + f_{vi}^2) \quad (6)$$

$$Q_i = -k_i m_i E_{DC} \left(e_{vi} \sin j_i - B_{vi,j} \cos j_i \right) + G_{vi,j} (e_{vi}^2 + f_{vi}^2) \quad (7)$$

$$P_{0i} = -k_i m_i E_{DC} \left(e_{0i} \cos j_i + B_{0i,j} \sin j_i \right) + G_{0i,j} (e_{0i}^2 + f_{0i}^2) \quad (8)$$

$$Q_{0i} = -k_i m_i E_{DC} \left(e_{0i} \sin j_i - B_{0i,j} \cos j_i \right) + G_{0i,j} (e_{0i}^2 + f_{0i}^2) \quad (9)$$

$$P_{vi} = -k_i m_i E_{DC} \left(e_{vi} \cos j_i + B_{vi,j} \sin j_i \right) + G_{vi,j} (e_{vi}^2 + f_{vi}^2) \quad (10)$$

$$Q_{vi} = -k_i m_i E_{DC} \left(e_{vi} \sin j_i - B_{vi,j} \cos j_i \right) + G_{vi,j} (e_{vi}^2 + f_{vi}^2) \quad (11)$$

From the generic expressions (6)-(11), the nodal active and reactive powers: P_{v1} , Q_{v1} , P_{01} , Q_{01} , P_{v2} , Q_{v2} , P_{02} , Q_{02} , P_{v3} , Q_{v3} , P_{03} , Q_{03} , become readily available by simply replacing the subscript i by 1, 2 and 3. Furthermore, the DC power contributions, in explicit form, are:

$$\begin{aligned} P_{DC1}^d &= (G_{DC1} + G_{DC2}) E_{DC1}^2 - G_{DC1} E_{DC1} E_{DC2} - G_{DC2} E_{DC1} E_{DC3} \\ P_{DC2}^d &= (G_{DC1} + G_{DC3}) E_{DC2}^2 - G_{DC1} E_{DC2} E_{DC1} - G_{DC3} E_{DC2} E_{DC3} \\ P_{DC3}^d &= (G_{DC2} + G_{DC3}) E_{DC3}^2 - G_{DC2} E_{DC3} E_{DC1} - G_{DC3} E_{DC3} E_{DC2} \end{aligned} \quad (12)$$

(b) VSC types

Borrowing the concept used in conventional AC power flows, relating to the bus classification into three different types, namely, slack, PV and PQ, Table I introduces three types of VSC stations which are required to solve the generic DC power grid problem put forward in this paper.

TABLE I: Types of VSC and their Control Variables

Type	Known variables	Unknown variables
VSC _{Slack}	E_{DC} , V_v^{spec}	m_a , j , B_{eq} , e_v , f_v
VSC _{Psch}	P^{sch} , V_v^{spec}	E_{DC} , m_a , j , B_{eq} , e_v , f_v
VSC _{Pass}	j , V_v^{spec}	E_{DC} , m_a , B_{eq} , e_v , f_v

The slack converter VSC_{Slack} provides voltage control at its DC terminal and it is linked on its AC side to a network which contains synchronous generation; the converter of type VSC_{Psch} serves the purpose of injecting a scheduled power into the DC grid and it is also linked on its AC side to a network with synchronous generation; the third type of VSC station is the passive converter VSC_{Pass} which is used to interconnect the DC grid with an AC network which contains no synchronous generation of its own. In the passive AC power grids the VSC's internal angle, j , provides the angular reference for the network.

(c) Mismatch Powers

The non-linear equation set (6)-(13) is solved by iteration using the Newton-Raphson method. To this end, mismatch power equations are set up where upon convergence of the iterative solution, the mismatch between the specified powers and the calculated powers become smaller than a pre-specified tolerance, at every node of the power system.

$$\begin{aligned} DP_i &= P_i^{net} - P_i^{cal} \\ DQ_i &= Q_i^{net} - Q_i^{cal} \\ DP_{vi} &= P_{vi}^{net} - P_{vi}^{cal} \\ DQ_{vi} &= Q_{vi}^{net} - Q_{vi}^{cal} \\ DP_{0i} &= P_{0i}^{net} - (P_{0i} + P_{DCi}^{cal}) \\ DU_{vi} &= V_{vi}^{spec} - \sqrt{e_{vi}^2 + f_{vi}^2} \end{aligned} \quad (14)$$

The superscript *net* is used to signify the power difference between an external injection of power by a source connected at given node and a load connected at the same node.

Furthermore, a reactive power constraining equation is required for all three types of converters to prevent the flow of reactive power into the DC grid. In connection with the three-terminal network in Fig. 1, this reactive power constraining equation is:

$$DQ_{0i-vi} = 0 - Q_{0i} \quad (15)$$

Moreover, converters of type VSC_{Psch} require an active power constraining equation which for the three-terminal networks in Fig. 1, and with no loss of generality, takes the following form:

$$DP_{0i-vi} = P_{0i-vi}^{sch} - P_{0i} \quad (16)$$

where P_{0i-vi}^{sch} is the amount of DC power entering inverter i at its DC bus.

(d) Linearized system of equations

Assuming that VSC₁ operates as a rectifier and VSC₂ and VSC₃ operate as inverters then linearization of (14)-(16) around the following base operating point:

to regulate power on the DC bus of the rectifier and to regulate the voltage magnitude at both the rectifier AC bus and the inverters AC buses. The relevant system of linearized equations is arranged, using compact notation, in the structure shown in eqn. (17).

$$\begin{array}{ccccc} \hat{\mathbf{E}}_{\text{VSC},1}^{\text{F}} \hat{\mathbf{U}} & \hat{\mathbf{E}}_{11}^{\text{J}} & \mathbf{0} & \mathbf{0} & \mathbf{J}_{1\text{DC}} \hat{\mathbf{U}} \hat{\Delta\Phi}_{\text{VSC},1}^{\text{J}} \hat{\mathbf{U}} \\ \hat{\mathbf{E}}_{\text{VSC},2}^{\text{F}} \hat{\mathbf{U}} & \hat{\mathbf{E}}_{21}^{\text{J}} & \mathbf{0} & \mathbf{J}_{22} & \mathbf{0} & \mathbf{J}_{2\text{DC}} \hat{\mathbf{U}} \hat{\Delta\Phi}_{\text{VSC},2}^{\text{J}} \hat{\mathbf{U}} \\ \hat{\mathbf{E}}_{\text{VSC},3}^{\text{F}} \hat{\mathbf{U}} & \hat{\mathbf{E}}_{31}^{\text{J}} & \mathbf{0} & \mathbf{0} & \mathbf{J}_{33} & \mathbf{J}_{3\text{DC}} \hat{\mathbf{U}} \hat{\Delta\Phi}_{\text{VSC},3}^{\text{J}} \hat{\mathbf{U}} \\ \hat{\mathbf{E}}_{\text{DC}}^{\text{F}} \hat{\mathbf{U}} & \hat{\mathbf{E}}_{\text{DC1}}^{\text{J}} & \mathbf{J}_{\text{DC2}} & \mathbf{J}_{\text{DC3}} & \mathbf{J}_{\text{DC}} \hat{\mathbf{U}} \hat{\Delta\mathbf{E}}_{\text{DC}}^{\text{J}} \hat{\mathbf{U}} \end{array} \quad (17)$$

where the $\mathbf{0}$ entries are zero-padded matrices of suitable orders.

In addition to the matrix entries corresponding to the three VSCs, \mathbf{J}_{11} , \mathbf{J}_{22} , \mathbf{J}_{33} , there are matrix entries corresponding to the DC grid, \mathbf{J}_{DC} , and mutual matrix terms between the DC nodes and their respective AC nodes: \mathbf{J}_{1DC} , \mathbf{J}_{2DC} , \mathbf{J}_{3DC} , \mathbf{J}_{DC1} , \mathbf{J}_{DC2} , \mathbf{J}_{DC3} .

The matrix entries \mathbf{J}_{11} , \mathbf{J}_{22} , \mathbf{J}_{33} into the higher order matrix in (17), correspond to first order partial derivatives of the power and voltage mismatch vectors in (18)-(20) with respect to the state variables increments in (21)-(23):

$$\mathbf{F}_{\text{VSCI}} = [\mathbf{D}P_{v1} \quad \mathbf{D}Q_{v1} \quad \mathbf{D}U_{v1} \quad \mathbf{D}Q_{0|v1} \quad \mathbf{D}P_{v1}]^T \quad (18)$$

$$\mathbf{F}_{\text{VSC2}} = [\mathbf{D}P_{v2} \quad \mathbf{D}Q_{v2} \quad \mathbf{D}U_{v2} \quad \mathbf{D}Q_{02-v2} \quad \mathbf{D}P_{02-v2}]^t \quad (19)$$

$$\mathbf{F}_{\text{VSC3}} = [\mathbf{D}P_{v3} \quad \mathbf{D}Q_{v3} \quad \mathbf{D}U_{v3} \quad \mathbf{D}Q_{03-v3}]^T \quad (20)$$

$$\Delta \Phi_{\text{VSC1}} = \frac{1}{E} \left(D_{e_{v1}} D_{m_{a1}} D_{f_{v1}} D_{B_{eq1}} D_{j_1} \right) \dot{\theta}^t \quad (21)$$

$$\Delta \Phi_{\text{VSC2}} = \begin{bmatrix} D_{e_{v2}} & D_{m_{a2}} & D_{f_{v2}} & D_{B_{eq2}} & D_{j_2} \end{bmatrix} \dot{\theta}^t \quad (22)$$

$$\Delta\Phi_{\text{VSC3}} = \dot{\theta} D_{e_{v3}} D_{m_{a3}} D_{f_{v3}} D_{B_{eq3}} \dot{\theta}^t \quad (23)$$

The mutual matrix entries $\mathbf{J}_{1\text{DC}}$, $\mathbf{J}_{2\text{DC}}$, $\mathbf{J}_{3\text{DC}}$ between the AC nodes and their corresponding DC nodes, correspond to first order partial derivatives of the mismatch vectors in (18)-(20) with respect to the state variables increments in (24):

$$\Delta \mathbf{E}_{\text{DC}} = [\mathbf{DE}_{DC2} \quad \mathbf{DE}_{DC3}]^T \quad (24)$$

Note that VSC_1 is acting as slack VSC in the DC grid and, therefore, $DE_{DC1} = 0$.

The mutual matrix entries \mathbf{J}_{DC1} , \mathbf{J}_{DC2} , \mathbf{J}_{DC3} between the DC nodes and their corresponding AC nodes, correspond to first order partial derivatives of the mismatch vector in (25) with respect to the state variables increments in (21)-(23):

$$\mathbf{F}_{\text{DC}} = [\mathbf{D}P_{\text{DC}2} \quad \mathbf{D}P_{\text{DC}3}]^t \quad (25)$$

The entry \mathbf{J}_{DC} corresponding to the DC part of the system, contains the partial derivatives of (25) with respect to the DC voltages in (24). Note that since VSC₁ is acting as slack VSC in the DC grid the matrix $\mathbf{J}_{\text{DC}1}$ is $\mathbf{0}$ but not \mathbf{J}_{DC} .

If no voltage regulation is exerted at the AC bus of any of the VSCs then suitable changes take place in (18)-(20) and

the corresponding matrix entries in (17). More explicitly, since the state variable m_{ai} is charged with regulating the AC voltage at node vi and voltage no regulation is exerted then m_{ai} becomes a constant parameter.

Conversely, if voltage regulation takes place at any of the DC buses then the corresponding row and column are deleted from (24), (25) and in the mutual matrix entries in (17). It should be remarked that the voltage must be specified in at least one of the buses of the DC network. Such a node plays the role of reference node in the DC network and in this three-terminal VSC-HVDC example this role has been assigned to VSC₁ which is VSC_{Stack} type.

The increments of the state variables in vector (17), calculated at iteration (r), are used to update the state variables, as follows:

$$\begin{aligned} e_{v1}^{(r)} &= e_{v1}^{(r-1)} + \mathbf{D} e_{v1}^{(r)} \\ f_{v1}^{(r)} &= f_{v1}^{(r-1)} + \mathbf{D} f_{v1}^{(r-1)} \\ m_{a1}^{(r)} &= m_{a1}^{(r-1)} + \mathbf{D} m_{a1}^{(r-1)} \\ B_{eq1}^{(r)} &= B_{eq1}^{(r-1)} + \mathbf{D} B_{eq1}^{(r)} \\ j_1^{(r)} &= j_1^{(r-1)} + \mathbf{D} j_1^{(r)} \end{aligned} \quad (26)$$

Similar expressions exist for updating the state variables of the inverters contained in vector (22) and (23).

The updating of the DC voltages using vector (24) is carried out as follows:

$$\begin{aligned} E_{DC2}^{(r)} &= E_{DC2}^{(r-1)} + \mathbf{D}\mathbf{E}_{DC2}^{(r)} \\ E_{DC3}^{(r)} &= E_{DC3}^{(r-1)} + \mathbf{D}\mathbf{E}_{DC3}^{(r)} \end{aligned} \quad (27)$$

It should be noted that when all entries relating to VSC_3 are removed in (4)-(27) then the three-terminal VSC-HVDC model reduces neatly to the more particular case of the point-to-point VSC-HVDC link model, a case in which it is equivalent to the point-to-point VSC-HVDC link model. However, it should be remarked that the model put forward in this paper is general and handles the DC link in explicit form. It is precisely the explicit representation of all the DC nodes in the formulation of this paper that enables general multi-terminal AC/DC power flow solutions in a truly unified way.

IV. MULTI-TERMINAL VSC-HVDC MODEL

A straightforward expansion of the linearized structure in (17), to include n rectifying stations, m inverting stations and an arbitrary DC network, yields the following result:

[illegible]

In this expression, each one of the n and m terms \mathbf{F} with subscripts VSC,R and VSC,I are vectors of power mismatches corresponding to n rectifying stations and m

inverting stations with an assigned operational characteristic of the type: VSC_{Slack} , VSC_{Psch} or VSC_{Pass} , according to operational requirements. Likewise, the vectors \mathbf{DF} contain the corresponding incremental state variable terms. By the same token, matrices \mathbf{J} with subscripts RR and II contain first order partial derivatives of the rectifier and inverter stations' state variables. The higher order Jacobian terms with subscripts DCR, DCI, RDC and IDC are interfacing terms between the AC and DC sides of the converter stations' state variables. The terms with subscripts DC contain powers and voltages belonging to the state variables of the DC network.

Since the multi-terminal VSC-HVDC system is used to interconnect a number of otherwise independent micro-grids, the linearised form of the overall electric power system's equations at a given iteration (r), is:

$$\begin{bmatrix} \hat{\mathbf{e}}_{AC} \mathbf{F}_{AC} \hat{\mathbf{u}}^{(r)} \\ \hat{\mathbf{e}}_{VSC,R} \mathbf{F}_{VSC,R} \hat{\mathbf{u}}^{(r)} \\ \hat{\mathbf{e}}_{VSC,I} \mathbf{F}_{VSC,I} \hat{\mathbf{u}}^{(r)} \\ \hat{\mathbf{e}}_{DC} \mathbf{F}_{DC} \hat{\mathbf{u}}^{(r)} \end{bmatrix} = - \mathbf{J}_{AC/DC} \begin{bmatrix} \hat{\mathbf{e}}_{AC} \mathbf{D}\Psi_{AC} \hat{\mathbf{u}}^{(r)} \\ \hat{\mathbf{e}}_{VSC,R} \mathbf{D}\Phi_{VSC,R} \hat{\mathbf{u}}^{(r)} \\ \hat{\mathbf{e}}_{VSC,I} \mathbf{D}\Phi_{VSC,I} \hat{\mathbf{u}}^{(r)} \\ \hat{\mathbf{e}}_{DC} \mathbf{D}\mathbf{E}_{DC} \hat{\mathbf{u}}^{(r)} \end{bmatrix} \quad (29)$$

V. TEST CASE

The five-terminal VSC-HVDC network shown in Fig. 3 is used to illustrate the applicability of the new frame-of-reference. It represents an AC/DC grid comprising a DC ring that interconnects two distribution systems (DS), a battery energy storage system (BESS), two micro-grids (MG) and a DFIG-based wind farm. For the purpose of this illustrative test case and with no loss of generality, each distribution system comprises 17 nodes, 16 distribution feeders and a total system load of 2.3 MW. The BESS is set to inject 0.5 MW from the DC ring. Each micro-grid draws 5 MW and operates at 0.9 lagging power factor. The wind farm contains four doubly-fed induction generators operating at its nominal power of 2 MW, to give an aggregated power of 8 MW.

Table II gives the parameters in per-unit values of the VSCs, DC ring cables, distribution lines of DS₁, DS₂ and WF.

TABLE II: Parameters of VSCs, DS₁, DS₂, WF and DC network

VSCs and LTCs						
G_0	Z_1	Y_{filter}	$Z_{reactor}$	Z_{LTC}		
$2e^{-4}$	$0.02+j0.10$	$j0.001$	$j0.10$	$j0.5$		
Distribution lines of DS ₁ , DS ₂ and WF						
Network	DS ₁	DS ₂		WF		
$R+jX$	$0.12+j0.20$	$0.12+j0.20$		$0.10+j0.25$		
Length (km) and resistance of the DC ring sections						
	$a-b$	$b-c$	$c-d$	$d-e$	$e-f$	$f-a$
Length	10	10	7.5	7.5	10	7.5
R_{DC}	0.5	0.5	0.375	0.375	0.50	0.375

Concerning the multi-terminal VSC-HVDC system, the LTC transformers of converters VSC₁ and VSC₂ are set to regulate voltage magnitudes at their high-voltage buses at 1 p.u. whereas the LTCs of converters VSC₃, VSC₄ and VSC₅ are set to exert voltage control at their high-voltage buses at 1.05 p.u. The station VSC₁ is selected to be type VSC_{Slack} providing voltage regulation at its DC bus at 1 p.u. It is assumed that the distribution system 2 imports 2.5 MW from the DC ring and, consequently, VSC₂ is set to be a converter of type VSC_{Psch} . The converters that link the two AC micro-grids and the wind farm with the DC ring, namely, VSC₃, VSC₄ and VSC₅ are modeled as converters of type VSC_{Pass} . The cables making up the DC ring are taken to be rated at ± 10 kV and having a resistance of 0.2 W/km^{-1} . The nominal powers of the converters are 10 MVA.

A sample of relevant power flow simulation results is provided in Table III, Table IV and Table V. As expected, convergence to a power mismatch tolerances of 10^{-6} and 10^{-12} are reached in 5 and 6 iterations, respectively, being this the hallmark of a true unified solution using the Newton-Raphson method.

TABLE III: State variable solution for each VSC

VSC type	E_{DC} (p.u.)	m_a	ϕ (deg)	LTC tap	P_{loss} (MW)
VSC _{Slack}	1.0000	0.8721	-13.3660	1.0354	0.0718
VSC _{Psch}	0.9952	0.8151	3.7648	0.9970	0.0066
VSC _{Pass}	0.9955	0.8842	0	1.0113	0.0281
VSC _{Pass}	1.0006	0.8612	0	1.0014	0.0571
VSC _{Pass}	0.9976	0.8824	0	1.0113	0.0282

Convergence: $\epsilon = 10^{-6}$ takes 5 iterations and $\epsilon = 10^{-12}$ takes 6 iterations

TABLE IV: Voltages and powers injected at the AC Networks

VSC	Network	V (p.u.)	θ (deg)	P_{inj} (MW)	Q_{inj} (MVar)
1	DS ₁	1.00	-9.5574	-4.3374	7.0417
2	DS ₂	1.00	1.4589	2.4933	-0.5987
3	MG ₁	1.05	-4.0565	5.0149	2.4448
4	WF	1.05	-6.7550	-7.9286	0.1782
5	MG ₂	1.05	-4.0565	5.0149	2.4448

TABLE V: Power flows in the DC ring

DC cables		P_{DC} (MW)		P_{loss} (kW)
		send-rec	rec-send	
a	b	1.6616	-1.6582	3.4514
b	c	2.1582	-2.1523	5.8467
c	d	-0.3476	0.3477	0.1143
d	e	-5.3908	5.4183	27.4889
e	f	2.4531	-2.4456	7.5132
f	a	-2.5975	2.6038	6.3563

From Table III, it is seen that converter VSC₄ injects power into the DC ring – its DC voltage is higher than the reference voltage provided by VSC₁, which is the slack VSC.

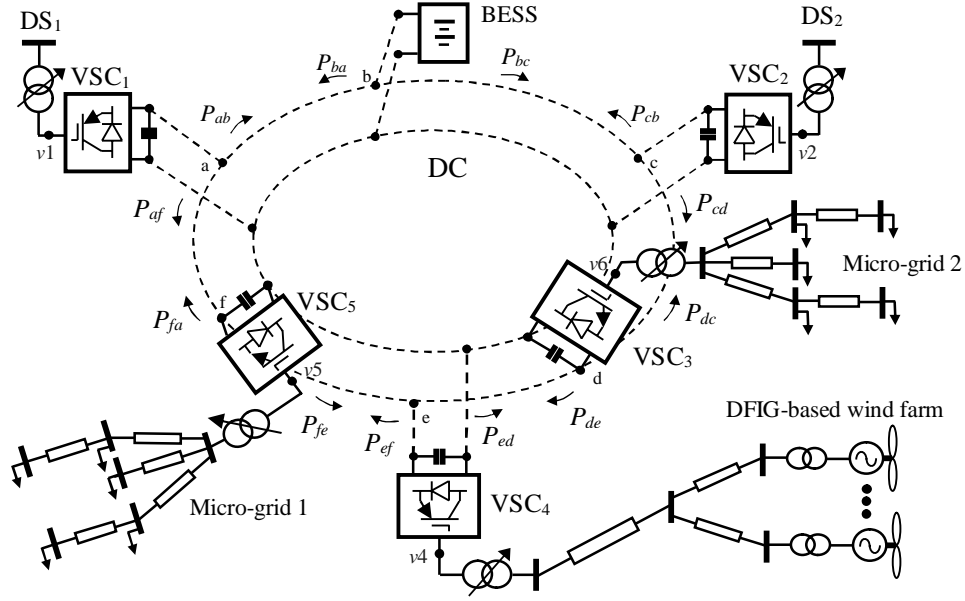


Figure 3. Multi-terminal VSC-HVDC system with micro-grids

The BESS injects 0.5 MW into the DC ring and its corresponding DC voltage is 0.9976 p.u. The angle j of converters VSC₃, VSC₄ and VSC₅ take values of zero since these converters provide the angular references for networks MG₁, MG₂ and WF. As shown in Table IV, the voltage phase angles of these nodes are displaced by -4.0565° , -4.0565° and -6.5750° , respectively. Notice that their amplitude modulation indexes take different values from each other since their voltage set points and reactive power injections are different.

The power loss incurred by each VSC is given in Table III. Converter VSC₁ incurs the highest loss and converter VSC₂ the lowest since it draws the lowest amount of power from the DC ring. The nodal voltages and active and reactive powers injected at the terminal of each AC network are given in Table IV. It should be noted that the LTC of VSC₁ injects 7.0417 MVar and the LTC of VSC₂ draws 0.5987 MVar in order to uphold the respective target voltages of DS₁ and DS₂ at 1 p.u. Also, VSC₂ is set to draw 2.5 MW from the DC ring and the power flow through VSC₁, from DS₁ and towards the DC ring, is 4.3374 MW.

The power control of a converter type VSC_{psch} applies at its DC bus; hence, the power delivered at its AC terminal will be slightly less due to the power loss incurred within the converter. In this test case, the power delivered to DS₂ stands at 2.4933 MW. Also, it should be noticed that the power injected by the LTCs at DS₁ and WF carry a negative sign which correctly accounts for the fact that the powers are being injected into the DC ring. The opposite occurs for the case of DS₂, MG₁ and MG₂, which draw power from the DC grid.

The power flows in the DC ring are given in Table V. It is noticed that the heavily loaded ring sector is $d-e$, which carries 5.3908 MW. Conversely, the less loaded is sector $c-d$, since most of the power imported by DS₂ (VSC₂) comes from the branches connecting VSC₁ and the EV charging station. The total power loss in the DC ring stand at 50.77 kW.

VI. CONCLUSIONS

A generalized frame-of-reference for the unified power flow solution of hybrid power systems has been presented and applied to the solution of low-power AC-DC grids (i.e., micro-grids). The new frame-of-reference is a linearised representation of the whole AC and DC power network around a base operating point enabling its iterative solution by means of the Newton-Raphson method, with a quadratic rate of convergence. The modeling flexibility of this computational framework enables de representation of a wide range of micro-grids, e.g. AC, DC and hybrid.

REFERENCES

- [1] S. Chowdhury, S.P. Chowdhury and P. Crossley, *Microgrids and Active Distribution Networks*, IET Renewable Energy Series 6, 2009.
- [2] Y. Ekanayake, N. Jenkins, N., K. Liyanage, J. Wu and A. Yokoyama: *Smart Grid: Technology and Applications*. Wiley & Sons, 2010.
- [3] European Commission – Directorate-General for Research, “Sustainable Energy Systems: European Smart Grids Technology Platform”, 2006.
- [4] Strzelecki, R.M. (Ed.): *Power Electronics in Smart Electrical Energy Networks*. Springer, 2008.
- [5] Carbone, R. (Ed.): *Energy Storage in the Emerging Era of Smart Grids*. INTECH, 2012.
- [6] Beerten, J., Cole S. and Belmans, R.: “Modelling of Multi-Terminal VSC HVDC Systems with Distributed DC Voltage Control”, IEEE Trans. on Power Systems, vol. 29, no. 1, 2014, pp. 34-42.
- [7] M. Baradar, and M. Ghandhari, “A Multi-Option Unified Power Flow Approach for Hybrid AC/DC Grids Incorporating Multi-Terminal VSC-HVDC”, IEEE Trans. on Power Systems, vol. 28, no. 3, Aug. 2013, pp. 2376-2383.
- [8] R. Chai, B. Zhang, J. Dou, Z. Hao and T. Zheng, “Unified Power Flow Algorithm Based on the NR Method for Hybrid AC/DC Grids Incorporating VSCs”, IEEE Trans. on Power Systems, DOI 10.1109/TPWRS.2015.2511303.
- [9] E. Acha and B. Kazemtabrizi, “A New STATCOM Model for Power Flows Using the Newton-Raphson Method”, IEEE Trans. on Power Systems, vol. 28, no. 3, Aug 2013, pp. 2455-2465.
- [10] E. Acha, *Nodal Analysis & Sparsity*, ADP-London, Kindle Publishing, Feb. 2015.

APPENDIX B. THE POWER FLOW PROGRAM

PF3PH-VSC-Start.m

```
%
% *****
%      *   THREE PHASE POWER FLOWS WITH VSC   *
%      *                   MAIN PROGRAM                   *
% *****
clear;clc;

% Import Data
PFData=Microgrid;
VSCData=Microgrid;

% Initial Calculations
[YR,YI] = YBus3Ph(nbb,ntl,tlsend,tlrec,TLImpedInv,TLAdmit,ntf,tfsend,tfrec,
    tftap,TFAdmit,nsh,shbus,shresis,shreac);

% Newton-Raphson Methode
[VM,VA,mVSC,phiVSC,BVSC0,it,PCAL,QCAL,VSCPCAL,VSCQCAL] = VSCNewtonRaphson3Ph
    (nmax,tol,itmax,ntl,ngn,nld,nbb,bustype,genbus,loadbus,tlsend,tlrec,
    TLImpedInv,TLAdmit,PGEN,QGEN,QMAX,QMIN,PLOAD,QLOAD,YR,YI,VM,VA,NVSC,
    VSCsend,VSCrec,RVSC1,XVSC1,GVSC0,BVSC0,BVSC0Lo,BVSC0Hi,mVSCHi,mVSCLo,mVSC
    ,phiVSC,VSCVMT,VSCPF,VSCC,VSCGCtrl,VSCVCtrl,VSCPCtrl,VSCtype);

% Calculation of Final power flow
[PCAL,QCAL] = CalculatedPowers3Ph(nbb,VM,VA,YR,YI);
[PCAL,QCAL,VSCPCAL,VSCQCAL] = VSCCalculatedPowers3Ph(VM,VA,PCAL,QCAL,NVSC,
    VSCsend,VSCrec,RVSC1,XVSC1,GVSC0,BVSC0,mVSC,phiVSC,VSCtype);
[PQsend,PQrec,PQloss,PQbus] = PQflows3Ph(nbb,ngn,ntl,nld,genbus,loadbus,
    tlsend,tlrec,PLOAD,QLOAD,VM,VA,TLImpedInv,TLAdmit);
[PQbus,PQVSCsend,PQVSCrec,PQVSC0,PQVSCLoss,BVSC0] = VSCPQflows3Ph(VM,VA,
    PQbus,NVSC,VSCsend,VSCrec,RVSC1,XVSC1,GVSC0,BVSC0,mVSC,phiVSC,VSCtype);
```

```

% Final Results
'Final Results (VA in Degrees)'
it                               %Iteration number
VM                               %Nodal voltage magnitude (p.u.)
VA = VA*180/pi                  %Nodal voltage phase angle(Deg)
mVSC                             %VSC voltage modulation ratio
phi = phiVSC*180/pi             %Final transformer phase-shifting position
BVSC0                           %Equivalent VSC Susceptance
PQbus                            %Node active and reactive power (p.u.)
PQsend                           %Sending active and reactive powers (p.u.)
PQrec                            %Receiving active and reactive powers (p.u.)
PQVSCsend                       %Sending SVC active and reactive powers (p.u.)
PQVSCrec                        %Receiving VSC active and reactive powers (p.u.)
PQVSC0                          %VSC active and reactive powers (p.u.)
PQloss                           %Active and reactive power losses (p.u.)
PQVSCLoss                       %VSC active and reactive power losses (p.u.)

save('measurements.mat','VM','mVSC','PCAL','QCAL','VSCPCAL','VSCQCAL','
    PQsend','PQrec','PQVSCsend','PQVSCrec','NVSC','VSCsend','VSCrec','
    VSCPCtrl')

% End of Main VSCPowerFlow3Ph Program

%      Written by Tom Rubbrecht, Tampere 2016
%      Inspired by Positive Sequence Power Flow STATCOM model – Enrique
%      Acha and Behzad Kazemtabrizi

```

PFData-Microgrid.m

```

% *****
%      *      THREE PHASE POWER FLOWS      *
%      *      TEST DATA 3Ph      *
% *****

% Busbars data
%bustype 1 :slack bus (VM,VA)
%bustype 2 :generator or PV bus
%bustype 3 :load or PQ bus
nbb=23;
bustype(1)=1;
    VM(1,1)=1.05; VA(1,1)=0;...
    VM(1,2)=1.05; VA(1,2)=-120*pi/180;...
    VM(1,3)=1.05; VA(1,3)=120*pi/180;
bustype(2)=3;
    VM(2,1)=1.00; VA(2,1)=0;...
    VM(2,2)=1.00; VA(2,2)=-120*pi/180;...
    VM(2,3)=1.00; VA(2,3)=120*pi/180;

```

```

bustype(3)=3;
    VM(3,1)=1.00; VA(3,1)=0;...
    VM(3,2)=1.00; VA(3,2)=-120*pi/180;...
    VM(3,3)=1.00; VA(3,3)=120*pi/180;
bustype(4)=3;
    VM(4,1)=1.00; VA(4,1)=0;...
    VM(4,2)=1.00; VA(4,2)=-120*pi/180;...
    VM(4,3)=1.00; VA(4,3)=120*pi/180;
bustype(5)=3;
    VM(5,1)=1.00; VA(5,1)=0;...
    VM(5,2)=1.00; VA(5,2)=-120*pi/180;...
    VM(5,3)=1.00; VA(5,3)=120*pi/180;
bustype(6)=3;
    VM(6,1)=1.00; VA(6,1)=0;...
    VM(6,2)=1.00; VA(6,2)=-120*pi/180;...
    VM(6,3)=1.00; VA(6,3)=120*pi/180;
bustype(7)=3;
    VM(7,1)=1.00; VA(7,1)=0;...
    VM(7,2)=1.00; VA(7,2)=-120*pi/180;...
    VM(7,3)=1.00; VA(7,3)=120*pi/180;
bustype(8)=3;
    VM(8,1)=1.00; VA(8,1)=0;...
    VM(8,2)=1.00; VA(8,2)=-120*pi/180;...
    VM(8,3)=1.00; VA(8,3)=120*pi/180;
bustype(9)=3;
    VM(9,1)=1.00; VA(9,1)=0;...
    VM(9,2)=1.00; VA(9,2)=-120*pi/180;...
    VM(9,3)=1.00; VA(9,3)=120*pi/180;
bustype(10)=3;
    VM(10,1)=1.00; VA(10,1)=0;...
    VM(10,2)=1.00; VA(10,2)=-120*pi/180;...
    VM(10,3)=1.00; VA(10,3)=120*pi/180;
bustype(11)=3;
    VM(11,1)=1.00; VA(11,1)=0;...
    VM(11,2)=1.00; VA(11,2)=-120*pi/180;...
    VM(11,3)=1.00; VA(11,3)=120*pi/180;
bustype(12)=3;
    VM(12,1)=1.00; VA(12,1)=0;...
    VM(12,2)=1.00; VA(12,2)=-120*pi/180;...
    VM(12,3)=1.00; VA(12,3)=120*pi/180;
bustype(13)=3;
    VM(13,1)=1.00; VA(13,1)=0;...
    VM(13,2)=1.00; VA(13,2)=-120*pi/180;...
    VM(13,3)=1.00; VA(13,3)=120*pi/180;
bustype(14)=3;
    VM(14,1)=1.00; VA(14,1)=0;...
    VM(14,2)=1.00; VA(14,2)=-120*pi/180;...
    VM(14,3)=1.00; VA(14,3)=120*pi/180;
bustype(15)=3;
    VM(15,1)=1.00; VA(15,1)=0;...
    VM(15,2)=1.00; VA(15,2)=-120*pi/180;...
    VM(15,3)=1.00; VA(15,3)=120*pi/180;

```

```

bustype(16)=3;
    VM(16,1)=1.00; VA(16,1)=0;...
    VM(16,2)=1.00; VA(16,2)=-120*pi/180;...
    VM(16,3)=1.00; VA(16,3)=120*pi/180;
bustype(17)=3;
    VM(17,1)=2.00; VA(17,1)=0;...
    VM(17,2)=2.00; VA(17,2)=0;...
    VM(17,3)=2.00; VA(17,3)=0;
bustype(18)=3;
    VM(18,1)=2.00; VA(18,1)=0;...
    VM(18,2)=2.00; VA(18,2)=0;...
    VM(18,3)=2.00; VA(18,3)=0;
bustype(19)=3;
    VM(19,1)=2.00; VA(19,1)=0;...
    VM(19,2)=2.00; VA(19,2)=0;...
    VM(19,3)=2.00; VA(19,3)=0;
bustype(20)=3;
    VM(20,1)=2.00; VA(20,1)=0;...
    VM(20,2)=2.00; VA(20,2)=0;...
    VM(20,3)=2.00; VA(20,3)=0;
bustype(21)=3;
    VM(21,1)=2.00; VA(21,1)=0;...
    VM(21,2)=2.00; VA(21,2)=0;...
    VM(21,3)=2.00; VA(21,3)=0;
bustype(22)=2;
    VM(22,1)=2.00; VA(22,1)=0;...
    VM(22,2)=2.00; VA(22,2)=0;...
    VM(22,3)=2.00; VA(22,3)=0;
bustype(23)=2;
    VM(23,1)=2.00; VA(23,1)=0;...
    VM(23,2)=2.00; VA(23,2)=0;...
    VM(23,3)=2.00; VA(23,3)=0;

% Transmission lines data
ntl=6;
tlsend(1)=2; tlrec(1)=3;
    lengthcable(1)=8;
tlsend(2)=2; tlrec(2)=5;
    lengthcable(2)=9;
tlsend(3)=3; tlrec(3)=8;
    lengthcable(3)=10;
tlsend(4)=5; tlrec(4)=14;
    lengthcable(4)=11;
tlsend(5)=8; tlrec(5)=11;
    lengthcable(5)=10;
tlsend(6)=11; tlrec(6)=14;
    lengthcable(6)=9;

```

```

Z_nonbonded=[0.1700+0.1653*i  0.0945-0.0127*i  0.0732-0.0286*i;...
              0.0945-0.0127*i  0.1583+0.1505*i  0.0945-0.0127*i;...
              0.0732-0.0286*i  0.0945-0.0127*i  0.1700+0.1653*i];
Y_reduced=10^-6*[0.0+71.99*i  0.0-3.22*i  0.0-1.51*i;...
                 0.0-3.22*i  0.0+72.10*i  0.0-3.22*i;...
                 0.0-1.51*i  0.0-3.22*i  0.0+71.99*i];

% Convert to per unit
Vref=132*10^3/sqrt(3);
Sref=100*10^6;
Zref=Vref^2/Sref;
Yref=1/Zref;

TLImpedInv = zeros(3,3,nt1);
TLAdmit = zeros(3,3,nt1);
for ii=1:nt1
    imped=Z_nonbonded*lengthcable(ii)/Zref;
    TLImpedInv(:,:,ii) = inv(imped);
    TLAdmit(:,:,ii)=Y_reduced*lengthcable(ii)/Yref;
end

% Transformers data
ntf=10;
tfsend(1)=1; tfrec(1)=2;
    tfresis(1)=0.012; tfreac(1)=0.120; tftap(1)=0.97;
tfsend(2)=3; tfrec(2)=4;
    tfresis(2)=0.030; tfreac(2)=0.300; tftap(2)=0.99;
tfsend(3)=5; tfrec(3)=6;
    tfresis(3)=0.015; tfreac(3)=0.150; tftap(3)=0.95;
tfsend(4)=5; tfrec(4)=7;
    tfresis(4)=0.030; tfreac(4)=0.300; tftap(4)=1.05;
tfsend(5)=8; tfrec(5)=9;
    tfresis(5)=0.015; tfreac(5)=0.150; tftap(5)=0.95;
tfsend(6)=8; tfrec(6)=10;
    tfresis(6)=0.030; tfreac(6)=0.300; tftap(6)=1.05;
tfsend(7)=11; tfrec(7)=12;
    tfresis(7)=0.030; tfreac(7)=0.300; tftap(7)=0.99;
tfsend(8)=11; tfrec(8)=13;
    tfresis(8)=0.030; tfreac(8)=0.300; tftap(8)=1.03;
tfsend(9)=14; tfrec(9)=15;
    tfresis(9)=0.030; tfreac(9)=0.300; tftap(9)=0.93;
tfsend(10)=14; tfrec(10)=16;
    tfresis(10)=0.030; tfreac(10)=0.300; tftap(10)=1.02;

TFAdmit=zeros(ntf,1);
for ii=1:ntf
    Z_tf=tfresis(ii)+tfreac(ii)*i;
    TFAdmit(ii)= inv(Z_tf);
end

```

% Generators data

```

ngn=3;
genbus(1)=1;
    PGEN(1,1)=0; QGEN(1,1)=0;
    PGEN(1,2)=0; QGEN(1,2)=0;
    PGEN(1,3)=0; QGEN(1,3)=0;
    QMAX(1)=9; QMIN(1)=-9;
genbus(2)=12;
    PGEN(2,1)=250/300; QGEN(2,1)=0.0;
    PGEN(2,2)=250/300; QGEN(2,2)=0.0;
    PGEN(2,3)=250/300; QGEN(2,3)=0.0;
    QMAX(2)=9; QMIN(2)=-9;
genbus(3)=23;
    PGEN(3,1)=50/300; QGEN(3,1)=0;
    PGEN(3,2)=50/300; QGEN(3,2)=0;
    PGEN(3,3)=50/300; QGEN(3,3)=0;
    QMAX(3)=9; QMIN(3)=-9;

```

% Loads data

```

nld=4;
loadbus(1)=22;
    PLOAD(1,1)=50/300;    QLOAD(1,1)=0.0;
    PLOAD(1,2)=50/300;    QLOAD(1,2)=0.0;
    PLOAD(1,3)=50/300;    QLOAD(1,3)=0.0;
loadbus(2)=6;
    PLOAD(2,1)=0.9*230.2/300;    QLOAD(2,1)=0.9*68.9/300;
    PLOAD(2,2)=230.2/300;        QLOAD(2,2)=68.9/300;
    PLOAD(2,3)=1.1*230.2/300;    QLOAD(2,3)=1.1*68.9/300;
loadbus(3)=9;
    PLOAD(3,1)=1.1*230.2/300;    QLOAD(3,1)=1.1*68.9/300;
    PLOAD(3,2)=230.2/300;        QLOAD(3,2)=68.9/300;
    PLOAD(3,3)=0.9*230.2/300;    QLOAD(3,3)=0.9*68.9/300;
loadbus(4)=15;
    PLOAD(4,1)=0.95*100/300;    QLOAD(4,1)=0.95*50/300;
    PLOAD(4,2)=1.05*100/300;    QLOAD(4,2)=1.05*50/300;
    PLOAD(4,3)=100/300;        QLOAD(4,3)=50/300;

```

% Shunts data

```

nsh = 0 ;
shbus(1) = 0 ;
shresis(1,1) = 0 ; shreac(1,1) = 0.0 ;
shresis(1,2) = 0 ; shreac(1,2) = 0.0 ;
shresis(1,3) = 0 ; shreac(1,3) = 0.0 ;

```

% General parameters

```

itmax=30;
tol=1e-12;
nmax=6*nbb;

```

% End of Data

VSCData-Microgrid.m

```

% *****
%          SINGLE PHASE VSC POWER FLOWS          *
%          VSC Data 3Bus network                  *
% *****

% Number of VSC's in network
NVSC = 7;
VSCtype=[1;1;1;1;1;2;2];

% Connectivity with network
VSCsend=[4;7;10;13;16;17;21];
VSCrec =[17;18;19;20;21;22;23];

% VSC Rating values
VSCP= [0.0;0.0;0.0;0.0;0.0;0.0;0.0]; VSCQ = [1.0;1.0;1.0;1.0;1.0;1.0;1.0];
Edc = [sqrt(2);sqrt(2);sqrt(2);sqrt(2);sqrt(2);sqrt(2);sqrt(2)];
VSCC= [sqrt(VSCP(1)^2+VSCQ(1)^2)/(Edc(1)/sqrt(2));...
       sqrt(VSCP(2)^2+VSCQ(2)^2)/(Edc(2)/sqrt(2));...
       sqrt(VSCP(3)^2+VSCQ(3)^2)/(Edc(3)/sqrt(2));...
       sqrt(VSCP(4)^2+VSCQ(4)^2)/(Edc(4)/sqrt(2));...
       sqrt(VSCP(5)^2+VSCQ(5)^2)/(Edc(5)/sqrt(2));...
       sqrt(VSCP(6)^2+VSCQ(6)^2)/(Edc(6)/sqrt(2));...
       sqrt(VSCP(7)^2+VSCQ(7)^2)/(Edc(7)/sqrt(2))];

% VSC impedance parameters
RVSC_01=0.01*3; XVSC_01=0.10*3; GVSC_01=0.01/3; BVSC_01=0.50/3;
RVSC_02=0.01*3; XVSC_02=0.10*3; GVSC_02=0.001/3; BVSC_02=0.50/3;

RVSC1=[RVSC_01 RVSC_01 RVSC_01;...
       RVSC_01 RVSC_01 RVSC_01;...
       RVSC_01 RVSC_01 RVSC_01;...
       RVSC_01 RVSC_01 RVSC_01;...
       RVSC_02 RVSC_02 RVSC_02;...
       RVSC_02 RVSC_02 RVSC_02];
XVSC1=[XVSC_01 XVSC_01 XVSC_01;...
       XVSC_01 XVSC_01 XVSC_01;...
       XVSC_01 XVSC_01 XVSC_01;...
       XVSC_01 XVSC_01 XVSC_01;...
       XVSC_01 XVSC_01 XVSC_01;...
       XVSC_02 XVSC_02 XVSC_02;...
       XVSC_02 XVSC_02 XVSC_02];

```



```

GVSC0=[GVSC_01 GVSC_01 GVSC_01;...
        GVSC_01 GVSC_01 GVSC_01;...
        GVSC_01 GVSC_01 GVSC_01;...
        GVSC_01 GVSC_01 GVSC_01;...
        GVSC_01 GVSC_01 GVSC_01;...
        GVSC_02 GVSC_02 GVSC_02;...
        GVSC_02 GVSC_02 GVSC_02];
BVSC0=[BVSC_01 BVSC_01 BVSC_01;...
        BVSC_01 BVSC_01 BVSC_01;...
        BVSC_01 BVSC_01 BVSC_01;...
        BVSC_01 BVSC_01 BVSC_01;...
        BVSC_01 BVSC_01 BVSC_01;...
        BVSC_02 BVSC_02 BVSC_02;...
        BVSC_02 BVSC_02 BVSC_02];

% VSC Initial values
mVSC= [1.0 1.0 1.0;...
        1.0 1.0 1.0;...
        1.0 1.0 1.0;...
        1.0 1.0 1.0;...
        1.0 1.0 1.0;...
        1.0 1.0 1.0;...
        1.0 1.0 1.0];
phiVSC=[0 -120*pi/180 120*pi/180;...
        0 -120*pi/180 120*pi/180;...
        0 -120*pi/180 120*pi/180;...
        0 -120*pi/180 120*pi/180;...
        0 -120*pi/180 120*pi/180;...
        0 0*pi/180 0*pi/180;...
        0 0*pi/180 0*pi/180];

% VSC limits
mVSCHi = [5;5;5;5;5;99;99]; mVSCLo = [0.2;0.2;0.2;0.2;0.2;0.2;0.2];
BVSC0Lo= [0;0;0;0;0;0;0]; BVSC0Hi= [BVSC0(1);BVSC0(2);BVSC0(3);BVSC0(4);
        BVSC0(5);BVSC0(6);BVSC0(7)];

% VSC Control parameters
%VSCGCtrl is the global control
    % = 0 is no control
    % = 1 is voltage control
    % = 2 is power flow control
    % = 3 is both voltage and power flow control
VSCGCtrl = [0;0;0;0;0;0;0];
    %VSCVCtrl is the voltage control; 1= on sending end; 2= on receiving
    end
VSCVCtrl = [0;0;0;0;0;0;0]; VSCVMT = [1.05;1.05;1.05;1.05;1.05;1.05;1.05];
    %VSCPCtrl is the power flow control; 1= on sending end (not working
    due to fundamental problems); 2 = on receiving end
VSCPCtrl = [0;0;0;0;0;0;0]; VSCPF = [0.0;0.0;0.0;0.0;0.0;0.0;0.0];

clear RVSC_01 XVSC_01 GVSC_01 BVSC_01 RVSC_02 XVSC_02 GVSC_02 BVSC_02
% End of VSCData

```

YBus3Ph.m

```

function [YR,YI] = YBus3Ph(nbb, ntl, tlsend, tlrec, TLImpedInv, TLAdmit, ntf,
    tfsend, tfrec, tftap, TFAdmit, nsh, shbus, shresis, shreac)

YY = zeros(nbb*3,nbb*3);

% Transmission lines contribution
for kk = 1 : ntl
    ii = (tlsend(kk)-1)*3 + 1;
    jj = (tlrec(kk)-1)*3 + 1;
    YY(ii:ii+2,ii:ii+2) = YY(ii:ii+2,ii:ii+2) + TLImpedInv(:, :, kk) +
        0.5*TLAdmit(:, :, kk);
    YY(ii:ii+2,jj:jj+2) = YY(ii:ii+2,jj:jj+2) - TLImpedInv(:, :, kk);
    YY(jj:jj+2,ii:ii+2) = YY(jj:jj+2,ii:ii+2) - TLImpedInv(:, :, kk);
    YY(jj:jj+2,jj:jj+2) = YY(jj:jj+2,jj:jj+2) + TLImpedInv(:, :, kk) +
        0.5*TLAdmit(:, :, kk);
end

% Transformer contribution
for kk = 1: ntf
    ii = (tfsend(kk)-1)*3 + 1;
    jj = (tfrec(kk)-1)*3 + 1;
    for pp=1:3
        YY(ii+pp-1,ii+pp-1) = YY(ii+pp-1,ii+pp-1) + TFAdmit(kk);
        YY(ii+pp-1,jj+pp-1) = YY(ii+pp-1,jj+pp-1) - tftap(kk)*TFAdmit(kk);
        YY(jj+pp-1,ii+pp-1) = YY(jj+pp-1,ii+pp-1) - tftap(kk)*TFAdmit(kk);
        YY(jj+pp-1,jj+pp-1) = YY(jj+pp-1,jj+pp-1) + tftap(kk)^2*TFAdmit(kk);
    end
end

% Shunt elements contribution
for kk = 1 : nsh
    SHAdmit = zeros(3,3);
    jj = shbus(kk)*3;
    for ii = 1 : 3
        SHAdmit(ii,ii) = 1/(shresis(kk,ii) + shreac(kk,ii)*i);
    end
    YY(jj-2:jj, jj-2:jj) = YY(jj-2:jj, jj-2:jj) + SHAdmit(:, :);
end

YR = real(YY);
YI = imag(YY);
% End YBus3Ph

```

VSCNewtonRaphson3Ph.m

```

function [VM,VA,mVSC,phiVSC,BVSC0,it,PCAL,QCAL,VSCPCAL,VSCQCAL] =
    VSCNewtonRaphson3Ph(nmax,tol,itmax,ntl,ngn,nld,nbb,bustype,genbus,loadbus,
        tlsend,tlrec,TLImpedInv,TLAdmit,PGEN,QGEN,QMAX,QMIN,PLOAD,QLOAD,YR,YI,VM,
        VA,NVSC,VSCsend,VSCrec,RVSC1,XVSC1,GVSC0,BVSC0,BVSC0Lo,BVSC0Hi,mVSCHi,
        mVSCLo,mVSC,phiVSC,VSCVMT,VSCPF,VSCC,VSCGCtrl,VSCVCtrl,VSCPCtrl,VSCtype)

% GENERAL SETTINGS
flag = 0;
it = 1;

% CALCULATE NET POWERS
[PNET,QNET] = NetPowers3Ph(nbb,ngn,nld,genbus,loadbus,PGEN,QGEN,PLOAD,QLOAD);

while (it < itmax && flag==0)
    % CALCULATED POWERS
    [PCAL,QCAL] = CalculatedPowers3Ph(nbb,VM,VA,YR,YI);
    % CALCULATED STATCOM POWERS
    [PCAL,QCAL,VSCPCAL,VSCQCAL] = VSCCalculatedPowers3Ph(VM,VA,PCAL,QCAL,
        NVSC,VSCsend,VSCrec,RVSC1,XVSC1,GVSC0,BVSC0,mVSC,phiVSC,VSCtype);
    % CHECK FOR POWERS LIMITS
    [QNET,bustype] = GeneratorsLimits(ngn,genbus,bustype,QGEN,QMAX,QMIN,QCAL,
        QNET,nld,loadbus,QLOAD,it);
    % POWER MISMATCHES
    [DPQ,DP,DQ,flag] = PowerMismatches3Ph(nmax,nbb,tol,bustype,flag,PNET,
        QNET,PCAL,QCAL,NVSC,VSCsend,VSCrec,VSCPCAL,VSCQCAL);
    % STATCOM POWER MISMATCHES
    [DPQ,flag] = VSCPowerMismatches3Ph(nbb,DPQ,flag,tol,NVSC,VSCPF,VSCPCAL,
        VSCQCAL,VSCGCtrl,VSCPCtrl,VSCrec);
    %Check for convergence
    if flag == 1
        break
    end
    % JACOBIAN FORMATION
    [JAC] = NewtonRaphsonJacobian3Ph(nmax,nbb,PCAL,QCAL,VM,VA,YR,YI);
    % STATCOM JACOBIAN UPDATING
    [JAC] = VSCNewtonRaphsonJacobian3Ph(nbb,bustype,VM,VA,JAC,NVSC,VSCsend,
        VSCrec,mVSC,phiVSC,RVSC1,XVSC1,GVSC0,BVSC0,VSCPCAL,VSCQCAL,VSCGCtrl,
        VSCVCtrl,VSCPCtrl,VSCtype);
    % SOLVE FOR THE STATE VARIABLES VECTOR
    D = JAC\DPQ';
    % UPDATE STATE VARIABLES
    [VA,VM] = StateVariablesUpdates3Ph(nbb,D,VA,VM,NVSC,VSCrec);
    % UPDATE STATCOM TAPS
    [VM,VA,BVSC0,mVSC,phiVSC] = VSCUpdates3Ph(nbb,VM,VA,NVSC,VSCsend,VSCrec,
        BVSC0,VSCVMT,mVSC,phiVSC,VSCGCtrl,VSCVCtrl,VSCtype,D);
    % CHECK FOR POSSIBLE VSC TAPS LIMITS VIOLATIONS
    [mVSC,BVSC0,VSCGCtrl,VSCVCtrl] = VSCLimits3ph(NVSC,mVSC,mVSCHi,mVSCLo,
        BVSC0,BVSC0Lo,BVSC0Hi,VSCGCtrl,VSCVCtrl,it);
    it = it + 1;
end
% End VSCNewtonRaphson3ph

```

NetPowers3Ph.m

```

function [PNET,QNET] = NetPowers3Ph(nbb,ngn,nld,genbus,loadbus,PGEN,QGEN,
    PLOAD,QLOAD);
PNET = zeros(1,nbb*3);
QNET = zeros(1,nbb*3);
for ii = 1 : ngn
    for jj = 1 : 3
        PNET((genbus(ii)-1)*3 + jj) = PNET((genbus(ii)-1)*3 + jj) + ...
            PGEN(ii,jj);
        QNET((genbus(ii)-1)*3 + jj) = QNET((genbus(ii)-1)*3 + jj) + ...
            QGEN(ii,jj);
    end
end
for ii = 1 : nld
    for jj = 1 : 3
        PNET((loadbus(ii)-1)*3 + jj) = PNET((loadbus(ii)-1)*3 + jj) - ...
            PLOAD(ii,jj);
        QNET((loadbus(ii)-1)*3 + jj) = QNET((loadbus(ii)-1)*3 + jj) - ...
            QLOAD(ii,jj);
    end
end
% End NetPowers3Ph;

```

CalculatedPowers3Ph.m

```

function [PCAL,QCAL] = CalculatedPowers3Ph(nbb,VM,VA,YR,YI);
PCAL = zeros(1,3*nbb);
QCAL = zeros(1,3*nbb);
VV1=zeros(3*nbb,1);
YY=YR+YI*i;
for ii=1:nbb
    for jj=1:3
        iii=3*(ii-1)+jj;
        VV1(iii,1)= VM(ii,jj)*cos(VA(ii,jj))+i*VM(ii,jj)*sin(VA(ii,jj));
    end
end
II=YY*VV1;
for ii=1:3*nbb
    VV(ii,ii)=VV1(ii,1);
end
PP=VV*conj(II);
PCAL=real(PP. ');
QCAL=imag(PP. ');
% End CalculatedPowers3Ph

```

VSCCalculatedPowers3Ph.m

```

function [PCAL,QCAL,VSCPCAL,VSCQCAL] = VSCCalculatedPowers3Ph(VM,VA,PCAL,
    QCAL,NVSC, VSCsend, VSCrec, RVSC1, XVSC1, GVSC0, BVSC0, mVSC, phiVSC, VSCtype)
VSCPCAL = zeros(1,6*NVSC);
VSCQCAL = zeros(1,6*NVSC);
% Calculate VSC admittances
for ii = 1:NVSC
    if VSCtype(ii)==1
        k1 = sqrt(3/8);
    elseif VSCtype(ii)==2
        k1 = 1;
    end
    for jj=1:3
        D = RVSC1(ii,jj)^2 + XVSC1(ii,jj)^2 ;
        if D<1e-9
            YR1 = 0;
            YI1 = 0;
        else
            YR1 = RVSC1(ii,jj)/D;
            YI1 = -XVSC1(ii,jj)/D;
        end

        YR0 = GVSC0(ii,jj);
        YI0 = BVSC0(ii,jj);

% Calculate STATCOM powers
        kk = 6*(ii-1)+jj;
        VSCPCAL(kk) = YR1*VM(VSCsend(ii),jj)^2 - k1*mVSC(ii,jj)*VM(VSCsend(ii),
            jj)*VM(VSCrec(ii),jj)*(YR1*cos(VA(VSCsend(ii),jj)-VA(VSCrec(ii),jj)-
            phiVSC(ii,jj))+YI1*sin(VA(VSCsend(ii),jj)-VA(VSCrec(ii),jj)-phiVSC(ii,
            jj))) ;
        VSCQCAL(kk) = -YI1*VM(VSCsend(ii),jj)^2 - k1*mVSC(ii,jj)*VM(VSCsend(ii),
            jj)*VM(VSCrec(ii),jj)*(YR1*sin(VA(VSCsend(ii),jj)-VA(VSCrec(ii),jj)-
            phiVSC(ii,jj))-YI1*cos(VA(VSCsend(ii),jj)-VA(VSCrec(ii),jj)-phiVSC(ii,
            jj))) ;
        VSCPCAL(kk+3) = (YR0+YR1*(k1*mVSC(ii,jj))^2)*VM(VSCrec(ii),jj)^2 - k1*
            mVSC(ii,jj)*VM(VSCsend(ii),jj)*VM(VSCrec(ii),jj)*(YR1*cos(VA(VSCrec(
            ii),jj)-VA(VSCsend(ii),jj)+phiVSC(ii,jj))+YI1*sin(VA(VSCrec(ii),jj)-
            VA(VSCsend(ii),jj)+phiVSC(ii,jj))) ;
        VSCQCAL(kk+3) = -((YI0+YI1)*(k1*mVSC(ii,jj))^2)*VM(VSCrec(ii),jj)^2 - k1
            *mVSC(ii,jj)*VM(VSCsend(ii),jj)*VM(VSCrec(ii),jj)*(YR1*sin(VA(VSCrec(
            ii),jj)-VA(VSCsend(ii),jj)+phiVSC(ii,jj))-YI1*cos(VA(VSCrec(ii),jj)-
            VA(VSCsend(ii),jj)+phiVSC(ii,jj))) ;
    end
end
% End VSCCalculatedPowers3Ph

```

GeneratorsLimits.m

```

function [QNET,bustype] = GeneratorsLimits(ngn,genbus,bustype,QGEN,QMAX,QMIN
,QCAL,QNET,nld,loadbus,QLOAD,it)
% CHECK FOR POSSIBLE GENERATOR'S REACTIVE POWERS LIMITS VIOLATIONS
if it > 2
    QCHECK = QCAL;
    for ii = 1 : nld
        for jj = 1:3
            QCHECK((loadbus(ii)-1)*3+jj)=QCHECK((loadbus(ii)-1)*3+jj)+QLOAD(ii ,
                jj);
        end
    end
    for ii = 1 : ngn
        jj = genbus(ii);
        for jjj = 1:3
            if ( bustype(jj) == 2 )
                if ( QCHECK((genbus(ii)-1)*3+jjj) > QMAX(ii) )
                    QNET((genbus(ii)-1)*3+jjj) = QMAX(ii);
                    bustype(jj) = 999;
                elseif ( QCHECK((genbus(ii)-1)*3+jjj) < QMIN(ii) )
                    QNET((genbus(ii)-1)*3+jjj) = QMIN(ii);
                    bustype(jj) = 999;
                end
            end
        end
    end
end
%End function Generatorslimits

```

PowerMismatches3Ph.m

```

function [DPQ,DP,DQ,flag] = PowerMismatches3Ph(nmax,nbb,tol,bustype,flag ,
    PNET,QNET,PCAL,QCAL,NVSC,VSCsend,VSCrec,VSCPCAL,VSCQCAL)
DPQ = zeros(1,nmax);
DP = zeros(1,nbb);
DQ = zeros(1,nbb);
DP = PNET - PCAL;
DQ = QNET - QCAL;

```

```

for ii = 1 : NVSC
    for jj=1:3
        aaa = (VSCsend(ii)-1)*3+jj; bbb = (VSCrec(ii)-1)*3+jj;
        kk = ((ii-1)*6+jj);
        DP(aaa) = DP(aaa) - VSCPCAL(kk);
        DQ(aaa) = DQ(aaa) - VSCQCAL(kk);
        DP(bbb) = DP(bbb) - VSCPCAL(kk+3);
        DQ(bbb) = DQ(bbb) - VSCQCAL(kk+3);
    end
end

% Remove active and reactive powers contributions of the slack bus and
% reactive power of all PV buses
kk = 1;
for ii = 1 : nbb
    for jj = 1 : 3
        if (bustype(ii) == 1 )
            DP(kk) = 0;
            DQ(kk) = 0;
        elseif (bustype(ii) == 2 )
            DQ(kk) = 0;
        end
        kk = kk + 1;
    end
end

% Re-arrange mismatch entries
kk = 1;
for ii = 1 : nbb
    for jj = 1 : 3
        DPQ((ii-1)*3 + kk) = DP(kk);
        DPQ((ii-1)*3 + kk + 3) = DQ(kk);
        kk = kk + 1;
    end
end

% Check for convergence
for ii = 1 : length(DPQ)
    if (abs(DPQ)<tol)
        flag = 1;
    end
end
%End function PowerMismatches3Ph

```

VSCPowerMismatches3Ph.m

```

function [DPQ, flag] = VSCPowerMismatches3Ph(nbb, DPQ, flag, tol, NVSC, VSCPF,
      VSCPCAL, VSCQCAL, VSCGCtrl, VSCPCtrl, VSCrec)
% ADD VSC POWER MISMATCHES TO DPQ
for ii = 1 : NVSC
    for jj = 1:3
        if VSCGCtrl(ii) <= 1
            DPQ(6*nbb+ii) = 0;
            DPQ(6*(nbb+ii)) = 0;
        else
            if VSCPCtrl(ii) == 1
                DPQ(6*nbb+4*(ii-1)+jj) = VSCPF(ii) - VSCPCAL(4*(ii-1)+jj);
            elseif VSCPCtrl(ii) == 2
                DPQ(6*nbb+4*(ii-1)+jj) = VSCPF(ii) + VSCPCAL(4*(ii-1)+3+jj);
            end
            DPQ(6*nbb+4*(ii-1)+3+jj) = 0 + VSCQCAL(4*(ii-1)+3+jj);
        end
    end
end

% Check for convergence
if (flag == 1)
    for ll = 6*nbb+1 : 6*(nbb+NVSC)
        if (abs(DPQ) < tol)
            flag = 1;
        else
            flag = 0;
        end
    end
end
%End function VSCPowerMismatches3Ph

```


NewtonRaphsonJacobian3Ph.m

```

function [JAC] = NewtonRaphsonJacobian3Ph(nmax,nbb,PCAL,QCAL,VM,VA,YR,YI)
% JACOBIAN FORMATION
JAC = zeros(nmax,nmax);
iii = 1;
for ii = 1 : nbb
    kk = (ii-1)*3 + 1;
    jjj = 1;
    for jj = 1 : nbb
        ll = (jj-1)*3 + 1;
        if ii == jj
            for mm=1:3;
                for nn=1:3;
                    if nn==mm
                        JAC(iii + mm-1,jjj + nn-1) = - QCAL(kk + mm-1) - VM(
                            ii ,mm) ...
                            ^2*YI(kk + mm-1,kk + mm-1);
                        JAC(iii + mm-1,3 + jjj + nn-1) = PCAL(kk + mm-1) +
                            ...
                            VM(ii ,mm)^2*YR(kk + mm-1,kk + mm-1);
                        JAC(iii + 3 + mm-1,jjj + nn-1) = PCAL(kk + mm-1) -
                            VM(ii ,mm) ...
                            ^2*YR(kk + mm-1,kk + mm-1);
                        JAC(iii + 3 + mm-1,jjj + 3 + nn-1) = QCAL(kk + mm-1)
                            - ...
                            VM(ii ,mm)^2*YI(kk + mm-1,kk + mm-1);
                    else
                        JAC(iii + mm-1,jjj + nn-1) = VM(ii ,mm)*VM(ii ,nn)*(YR
                            (kk + ...
                            mm-1,kk + nn-1)*sin(VA(ii ,mm)-VA(ii ,nn))-YI(kk +
                            mm-1,...
                            kk + nn-1)*cos(VA(ii ,mm)-VA(ii ,nn)));
                        JAC(iii + mm-1,3 + jjj + nn-1) = VM(ii ,mm)*VM(ii ,nn)
                            *...
                            (YR(kk + mm-1,kk + nn-1)*cos(VA(ii ,mm)-VA(ii ,nn)
                            ) + YI(kk + ...
                            mm-1,kk + nn-1)*sin(VA(ii ,mm)-VA(ii ,nn)));
                        JAC(iii + 3 + mm-1,jjj + nn-1) = -VM(ii ,mm)*VM(ii ,nn)
                            *...
                            (YR(kk + mm-1,kk + nn-1)*cos(VA(ii ,mm)-VA(ii ,nn)
                            ) + YI(kk + ...
                            mm-1,kk + nn-1)*sin(VA(ii ,mm)-VA(ii ,nn)));
                        JAC(iii + 3 + mm-1,jjj + 3 + nn-1) = VM(ii ,mm)*VM(ii
                            ,nn) ...
                            *(YR(kk + mm-1,kk + nn-1)*sin(VA(ii ,mm)-VA(ii ,nn)
                            ))-YI...
                            (kk + mm-1,kk + nn-1)*cos(VA(ii ,mm)-VA(ii ,nn)));
                    end
                end
            end
        end
    end
end

```

```

else
    for mm=1:3;
        for nn=1:3;
            JAC( iii + mm-1, jjj + nn-1) = VM(ii ,mm)*VM(jj ,nn)*(YR(kk +
                + ...
                mm-1,ll + nn-1)*sin(VA(ii ,mm)-VA(jj ,nn)) - YI(kk +
                mm-1,ll + ...
                nn-1)*cos(VA(ii ,mm)-VA(jj ,nn)));
            JAC( iii + mm-1,3 + jjj + nn-1) = VM(ii ,mm)*VM(jj ,nn)*(YR
                (kk + ...
                mm-1,ll + nn-1)*cos(VA(ii ,mm)-VA(jj ,nn)) + YI(kk +
                mm-1,ll + ...
                nn-1)*sin(VA(ii ,mm)-VA(jj ,nn)));
            JAC( iii + 3 + mm-1, jjj + nn-1) = -VM(ii ,mm)*VM(jj ,nn)*(
                YR(kk + ...
                mm-1,ll + nn-1)*cos(VA(ii ,mm)-VA(jj ,nn)) + YI(kk +
                mm-1,ll + ...
                nn-1)*sin(VA(ii ,mm)-VA(jj ,nn)));
            JAC( iii + 3 + mm-1, jjj + 3 + nn-1) = VM(ii ,mm)*VM(jj ,nn)
                ...
                *(YR(kk + mm-1,ll + nn-1)*sin(VA(ii ,mm)-VA(jj ,nn)) -
                YI(kk + ...
                mm-1,ll + nn-1)*cos(VA(ii ,mm)-VA(jj ,nn)));
        end
    end
    jjj = jjj + 6;
end
iii = iii + 6;
end
%End of function NewtonRaphsonJacobian3Ph

```

VSCNewtonRaphsonJacobian3Ph.m

```

function [JAC] = VSCNewtonRaphsonJacobian3Ph(nbb, bustype, VM, VA, JAC, NVSC,
    VSCsend, VSCrec, mVSC, phiVSC, RVSC1, XVSC1, GVSC0, BVSC0, VSCPCAL, VSCQCAL,
    VSCGCtrl, VSCVCtrl, VSCPCtrl, VSCtype)
%% VSC JACOBIAN MODIFICATION
JAC(6*nbb+NVSC, 1:6*(nbb+NVSC)) = zeros;
JAC(1:6*(nbb+NVSC), 6*nbb+NVSC) = zeros;
JC(1:3*NVSC, 6*(nbb+NVSC)) = zeros;
YR0=zeros(NVSC,3); YR1=zeros(NVSC,3);
YI0=zeros(NVSC,3); YI1=zeros(NVSC,3);

```

```

for ii = 1 : NVSC
    if VSCtype(ii)==1
        k1 = sqrt(3/8);
    elseif VSCtype(ii)==2
        k1 = 1;
    end

% Calculate VSC admittances
for jj=1:3
    D = RVSC1(ii,jj)^2 + XVSC1(ii,jj)^2;
    if D<1e-9
        YR1(ii,jj) = 0;
        YI1(ii,jj) = 0;
    else
        YR1(ii,jj) = RVSC1(ii,jj)/D;
        YI1(ii,jj) = -XVSC1(ii,jj)/D;
    end
    YR0(ii,jj) = GVSC0(ii,jj);
    YI0(ii,jj) = BVSC0(ii,jj);
end

% Calculate STATCOM Jacobian entries (Pvr & Qvr)
JKK= zeros(6,6);
JKM= zeros(6,6);
for jj=1:3
    nn1 = 4*(ii-1)+jj;
    for ll=1:3
        if jj==ll
            JKK(jj,ll)= -VSCQCAL(nn1) - YI1(ii,jj)*VM(VSCsend(ii),jj)^2+1e-12;
            JKK(jj,ll+3)= VSCPCAL(nn1) + YR1(ii,jj)*VM(VSCsend(ii),jj)^2;
            JKK(jj+3,ll)= VSCPCAL(nn1) - YR1(ii,jj)*VM(VSCsend(ii),jj)^2;
            JKK(jj+3,ll+3)= VSCQCAL(nn1) - YI1(ii,jj)*VM(VSCsend(ii),jj)^2+1e-12;

            JKM(jj,ll)= VSCQCAL(nn1) + YI1(ii,jj)*VM(VSCsend(ii),jj)^2;
            JKM(jj,ll+3)= VSCPCAL(nn1) - YR1(ii,jj)*VM(VSCsend(ii),jj)^2;
            JKM(jj+3,ll)= -VSCPCAL(nn1) + YR1(ii,jj)*VM(VSCsend(ii),jj)^2;
            JKM(jj+3,ll+3)= VSCQCAL(nn1) + YI1(ii,jj)*VM(VSCsend(ii),jj)^2;
        else
            JKK(jj,ll)= 0;
            JKK(jj,ll+3)= 0;
            JKK(jj+3,ll)= 0;
            JKK(jj+3,ll+3)= 0;

            JKM(jj,ll)= 0;
            JKM(jj,ll+3)= 0;
            JKM(jj+3,ll)= 0;
            JKM(jj+3,ll+3)= 0;
        end
    end
end
end

```

```

if (VSCGCtrl(ii)==1 || VSCGCtrl(ii)==3) %Voltage control or Voltage-Power
    flow control
    if (VSCVCtrl(ii) == 1) % primary control
        JAC(:,6*VSCsend(ii)-2:6*VSCsend(ii))= 0.0;
    elseif (VSCVCtrl(ii) == 2) % secondary control
        JAC(:,6*VSCrec(ii)-2:6*VSCrec(ii))= 0.0;
    end
    for jj=1:3
        nn1 = 4*(ii-1)+jj;
        for ll=1:3
            if (VSCVCtrl(ii)==1 && jj==ll) % primary control
                JKK(jj, ll+3)= VSCPCAL(nn1) - YR1(ii, jj)*VM(VSCsend(ii), jj)
                    ^2;
                JKK(jj+3, ll+3)= VSCQCAL(nn1) + YI1(ii, jj)*VM(VSCsend(ii), jj)
                    ^2;
            elseif (VSCVCtrl(ii)==2 && jj==ll) % secondary control
                JKM(jj, ll+3)= VSCPCAL(nn1) - YR1(ii, jj)*VM(VSCsend(ii), jj)
                    ^2;
                JKM(jj+3, ll+3)= VSCQCAL(nn1) + YI1(ii, jj)*VM(VSCsend(ii), jj)
                    ^2;
            end
        end
    end
end

% Add STATCOM contribution to system JAC
kk = 6*(VSCsend(ii)-1)+1;
mm = 6*(VSCrec(ii)-1)+1;
JAC(kk:kk+5, kk:kk+5) = JAC(kk:kk+5, kk:kk+5) + JKK;
JAC(kk:kk+5, mm:mm+5) = JAC(kk:kk+5, mm:mm+5) + JKM;

% Calculate VSC Jacobian entries (P0 % Q0)
JKK= zeros(6,6);
JKM= zeros(6,6);
for jj=1:3
    nn2 = 4*(ii-1)+jj+3;
    for ll=1:3
        if jj==ll
            JKK(jj, ll)= -VSCQCAL(nn2) - ((YI0(ii, jj)+YI1(ii, jj))*(k1*mVSC(ii,
                jj))^2)*VM(VSCrec(ii), jj)^2+1e-12;
            JKK(jj, ll+3)= VSCPCAL(nn2) + (YR0(ii, jj)+(k1*mVSC(ii, jj))^2*YR1(
                ii, jj))*VM(VSCrec(ii), jj)^2;
            JKK(jj+3, ll)= VSCPCAL(nn2) - (YR0(ii, jj)+(k1*mVSC(ii, jj))^2*YR1(
                ii, jj))*VM(VSCrec(ii), jj)^2;
            JKK(jj+3, ll+3)= VSCQCAL(nn2) - ((YI0(ii, jj)+YI1(ii, jj))*(k1*mVSC(
                ii, jj))^2)*VM(VSCrec(ii), jj)^2+1e-12;
        end
    end
end

```

```

JKM(jj, ll) = VSCQCAL(nn2) + ((YI0(ii, jj)+YI1(ii, jj))*(k1*mVSC(ii,
    , jj))^2)*VM(VSCrec(ii), jj)^2;
JKM(jj, ll+3) = VSCPCAL(nn2) - (YR0(ii, jj)+(k1*mVSC(ii, jj))^2*YR1
    (ii, jj))*VM(VSCrec(ii), jj)^2;
JKM(jj+3, ll) = -VSCPCAL(nn2) + (YR0(ii, jj)+(k1*mVSC(ii, jj))^2*
    YR1(ii, jj))*VM(VSCrec(ii), jj)^2;
JKM(jj+3, ll+3) = VSCQCAL(nn2) + ((YI0(ii, jj)+YI1(ii, jj))*(k1*
    mVSC(ii, jj))^2)*VM(VSCrec(ii), jj)^2;
else
    JKK(jj, ll)= 0;
    JKK(jj, ll+3)= 0;
    JKK(jj+3, ll)= 0;
    JKK(jj+3, ll+3)= 0;

    JKM(jj, ll) = 0;
    JKM(jj, ll+3) = 0;
    JKM(jj+3, ll) = 0;
    JKM(jj+3, ll+3) = 0;
end
end
end
for jj=1:3
    for ll=1:3
        if (VSCGCtrl(ii)==1 || VSCGCtrl(ii)==3) %Voltage control or
            Voltage-Power flow control
            if (VSCVCtrl(ii) == 1 && jj==ll) % primary control with m
                JKM(jj, ll+3) = VSCPCAL(nn2) + ((k1*mVSC(ii, jj))^2*YR1(ii,
                    , jj)-YR0(ii, jj))*VM(VSCrec(ii), jj)^2;
                JKM(jj+3, ll+3) = VSCQCAL(nn2) - ((k1*mVSC(ii, jj))^2*(YI1
                    (ii, jj)+YI0(ii, jj))*VM(VSCrec(ii), jj)^2;
            elseif (VSCVCtrl(ii) == 2 && jj==1) % secondary control with
                m
                JKK(jj, ll+3) = VSCPCAL(nn2) + ((k1*mVSC(ii, jj))^2*YR1(ii,
                    , jj) - YR0(ii, jj))*VM(VSCrec(ii), jj)^2;
                JKK(jj+3, ll+3) = VSCQCAL(nn2) - ((k1*mVSC(ii, jj))^2*(YI1
                    (ii, jj)+YI0(ii, jj))*VM(VSCrec(ii), jj)^2;
            end
        end
    end
end

% Add STATCOM contribution to system JAC
kk = 6*(VSCsend(ii)-1)+1;
mm = 6*(VSCrec(ii)-1)+1;
JAC(mm:mm+5, mm:mm+5) = JAC(mm:mm+5, mm:mm+5) + JKK;
JAC(mm:mm+5, kk:kk+5) = JAC(mm:mm+5, kk:kk+5) + JKM;

```

```

% Calculate VSC Jacobian entries for power regulation
if (VSCGCtrl(ii) <= 1)
    % Enter void values to row and column associated with active and
    % reactive power regulation
    pp = 6*(nbb+ii-1)+1;
    JAC(1:pp+2,pp:pp+2) = zeros;
    JAC(pp:pp+2,1:pp+2) = zeros;
    JAC(pp,pp) = 1; JAC(pp+1,pp+1) = 1; JAC(pp+2,pp+2) = 1;
    JAC(1:pp+5,pp+3:pp+5) = zeros;
    JAC(pp+3:pp+5,1:pp+5) = zeros;
    JAC(pp+3,pp+3) = 1; JAC(pp+4,pp+4) = 1; JAC(pp+5,pp+5) = 1;
else
    %Derivatives of active and reactive powers vs phase-shifter angle and
    % shunt susceptance
    JKE= zeros(6,6);
    JME= zeros(6,6);
    for jj= 1:3
        nn1 = 4*(ii-1)+jj;
        nn2 = 4*(ii-1)+jj+3;
        for ll=1:3
            if (jj==ll)
                JKE(jj, ll) = VSCQCAL(nn1) + YI1(ii, jj)*VM(VSCsend(ii), jj)^2;
                JKE(jj+3, ll) = -VSCPCAL(nn1) + YR1(ii, jj)*VM(VSCsend(ii), jj)^2;
            else
                JKE(jj, ll) = 0;
                JKE(jj+3, ll) = 0;
                JKE(jj, ll+3) = 0;
                JKE(jj+3, ll+3) = 0;
            end
            if (jj==ll)
                JME(jj, ll) = -VSCQCAL(nn2) - ((YI0(ii, jj)+YI1(ii, jj))*(k1*
                    mVSC(ii, jj))^2)*VM(VSCrec(ii), jj)^2;
                JME(jj, ll+3) = 0;
                JME(jj+3, ll) = -VSCPCAL(nn2) - (YR0(ii, jj)+YR1(ii, jj))*(k1*mVSC
                    (ii, jj))^2)*VM(VSCrec(ii), jj)^2;
                JME(jj+3, ll+3) = -(k1*mVSC(ii, ll))^2*VM(VSCrec(ii), jj)^2;
            else
                JME(jj, ll) = 0;
                JME(jj, ll+3) = 0;
                JME(jj+3, ll) = 0;
                JME(jj+3, ll+3) = 0;
            end
        end
    end
end
end

```

```

if VSCPCtrl(ii) == 1
    %Derivatives of regulated power mismatch vs state variables
    JEK= zeros(6,6);
    JEM= zeros(6,6);
    JE= zeros(6,6);
    for jj= 1:3
        nn1 = 4*(ii-1)+jj;
        nn2 = 4*(ii-1)+3+jj;
        for ll=1:3
            if jj==ll
                JEK(jj, ll) =-VSCQCAL(nn1) - YI1(ii, jj)*VM(VSCsend(ii), jj)
                    ^2;
                JEK(jj, ll+3) = VSCPCAL(nn1) + YR1(ii, jj)*VM(VSCsend(ii),
                    jj) ^2;
                JEK(jj+3, ll) =-VSCPCAL(nn2) + (YR0(ii, jj)+YR1(ii, jj)*(k1
                    *mVSC(ii, jj))^2)*VM(VSCrec(ii), jj) ^2;
                JEK(jj+3, ll+3) =-VSCQCAL(nn2) + ((YI0(ii, jj)+YI1(ii, jj))
                    *(k1*mVSC(ii, jj))^2)*VM(VSCrec(ii), jj) ^2;

                JEM(jj, ll) = VSCQCAL(nn1) + YI1(ii, jj)*VM(VSCsend(ii), jj)
                    ^2;
                JEM(jj, ll+3) = VSCPCAL(nn1) - YR1(ii, jj)*VM(VSCsend(ii),
                    jj) ^2;
                JEM(jj+3, ll) = VSCPCAL(nn2) - (YR0(ii, jj)+YR1(ii, jj)*(k1
                    *mVSC(ii, jj))^2)*VM(VSCrec(ii), jj) ^2;
                JEM(jj+3, ll+3) =-VSCQCAL(nn2) - ((YI0(ii, jj)+YI1(ii, jj))
                    *(k1*mVSC(ii, jj))^2)*VM(VSCrec(ii), jj) ^2;

                JE(jj, ll) = VSCQCAL(nn1) + YI1(ii, jj)*VM(VSCsend(ii), jj)
                    ^2;
                JE(jj, ll+3) = 0;
                JE(jj+3, ll) =-VSCPCAL(nn2) + (YR0(ii, jj)+(k1*mVSC(ii, jj)
                    )^2*YR1(ii, jj))*VM(VSCrec(ii), jj) ^2;
                JE(jj+3, ll+3) = (k1*mVSC(ii, jj))^2*VM(VSCrec(ii), jj) ^2;
            else
                JEK(jj, ll) = 0;
                JEK(jj, ll+3) = 0;
                JEK(jj+3, ll) = 0;
                JEK(jj+3, ll+3) = 0;

                JEM(jj, ll) = 0;
                JEM(jj, ll+3) = 0;
                JEM(jj+3, ll) = 0;
                JEM(jj+3, ll+3) = 0;

                JE(jj, ll) = 0;
                JE(jj, ll+3) = 0;
                JE(jj+3, ll) = 0;
                JE(jj+3, ll+3) = 0;
            end
        end
    end
end

```

```

elseif VSCPCtrl(ii) == 2
    %The power flow regulated at the secondary is Pkm=Pmk and the
    %derivatives below are taken accordingly
    %Derivatives of regulated power mismatch vs state variables
    JEK= zeros(6,6);
    JEM= zeros(6,6);
    JE= zeros(6,6);
    for jj= 1:3
        nn2 = 4*(ii-1)+3+jj;
        for ll=1:3
            if jj==ll
                JEK(jj, ll) =-VSCQCAL(nn2) - ((YI0(ii, jj)+YI1(ii, jj))*(k1
                    *mVSC(ii, jj))^2)*VM(VSCrec(ii), jj)^2;
                JEK(jj, ll+3) = -VSCPCAL(nn2) + (YR1(ii, jj)*(k1*mVSC(ii,
                    jj))^2+YR0(ii, jj))*VM(VSCrec(ii), jj)^2;
                JEK(jj+3, ll) = -VSCPCAL(nn2) + (YR1(ii, jj)*(k1*mVSC(ii,
                    jj)+YR0(ii, jj))^2)*VM(VSCrec(ii), jj)^2;
                JEK(jj+3, ll+3) =-VSCQCAL(nn2) + ((YI0(ii, jj)+YI1(ii, jj))
                    *(k1*mVSC(ii, jj))^2)*VM(VSCrec(ii), jj)^2;

                JEM(jj, ll) = VSCQCAL(nn2) + ((YI0(ii, jj)+YI1(ii, jj))*(k1
                    *mVSC(ii, jj))^2)*VM(VSCrec(ii), jj)^2;
                JEM(jj, ll+3) =-VSCPCAL(nn2) + (YR1(ii, jj)*(k1*mVSC(ii, jj)
                    )^2+YR0(ii, jj))*VM(VSCrec(ii), jj)^2;
                JEM(jj+3, ll) = VSCPCAL(nn2) + (YR1(ii, jj)*(k1*mVSC(ii, jj)
                    )^2+YR0(ii, jj))*VM(VSCrec(ii), jj)^2;
                JEM(jj+3, ll+3) =-VSCQCAL(nn2) - ((YI0(ii, jj)+YI1(ii, jj))
                    *(k1*mVSC(ii, jj))^2)*VM(VSCrec(ii), jj)^2;

                JE(jj, ll) = VSCQCAL(nn2) + ((YI0(ii, jj)+YI1(ii, jj))*(k1*
                    mVSC(ii, jj))^2)*VM(VSCrec(ii), jj)^2;
                JE(jj, ll+3) = 0;
                JE(jj+3, ll) =-VSCPCAL(nn2) + (YR0(ii, jj)+YR1(ii, jj))*(k1*
                    mVSC(ii, jj))^2)*VM(VSCrec(ii), jj)^2;
                JE(jj+3, ll+3) = (k1*mVSC(ii, jj))^2*VM(VSCrec(ii), jj)^2;
            else
                JEK(jj, ll) = 0;
                JEK(jj, ll+3) = 0;
                JEK(jj+3, ll) = 0;
                JEK(jj+3, ll+3) = 0;

                JEM(jj, ll) = 0;
                JEM(jj, ll+3) = 0;
                JEM(jj+3, ll) = 0;
                JEM(jj+3, ll+3) = 0;

                JE(jj, ll) = 0;
                JE(jj, ll+3) = 0;
                JE(jj+3, ll) = 0;
                JE(jj+3, ll+3) = 0;
            end
        end
    end
end
end

```



```

% Add VSC contribution to system JAC
kk = 6*(VSCsend(ii)-1)+1;
mm = 6*(VSCrec(ii)-1)+1;
pp = 6*(nbb+ii-1)+1;
JAC(kk:kk+5,pp:pp+5) = JAC(kk:kk+5,pp:pp+5) + JKE;
JAC(mm:mm+5,pp:pp+5) = JAC(mm:mm+5,pp:pp+5) + JME;
JAC(pp:pp+5,kk:kk+5) = JAC(pp:pp+5,kk:kk+5) + JEK;
JAC(pp:pp+5,mm:mm+5) = JAC(pp:pp+5,mm:mm+5) + JEM;
JAC(pp:pp+5,pp:pp+5) = JAC(pp:pp+5,pp:pp+5) + JE;

if (VSCGCtrl(ii)==3)
    for jj= 1:3
        pp = 6*(nbb+ii-1)+jj;
        nn1 = 4*(ii-1)+jj;
        for ll= 1:3
            kk = 6*(VSCsend(ii)-1)+ll+3;
            mm = 6*(VSCrec(ii)-1)+ll+3;

            if (VSCPCtrl(ii) == 1) && (VSCVCtrl(ii) == 1)
                %' primary power control & primary voltage control with
                m'
                if (jj==ll)
                    JAC(pp,kk) = VSCPCAL(nn1) - YR1(ii,jj)*VM(VSCsend(ii),jj)^2;
                else
                    JAC(pp,kk)=0;
                end
            elseif (VSCPCtrl(ii) == 1) && (VSCVCtrl(ii) == 2)
                %' primary power control & secondary voltage control
                with m'
                if (jj==ll)
                    JAC(pp,mm) = VSCPCAL(nn1) - YR1(ii,jj)*VM(VSCsend(ii),jj)^2;
                else
                    JAC(pp,mm)=0;
                end
            elseif (VSCPCtrl(ii) == 2) && (VSCVCtrl(ii)==1)
                %' secondary power control & primary voltage control
                with m'
                if (jj==ll)
                    JAC(pp,kk)= -VSCPCAL(nn2) - ((k1*mVSC(ii,jj))^2*YR1(ii,jj)-YR0(ii,jj))*VM(VSCrec(ii),jj)^2;
                else
                    JAC(pp,kk)=0;
                end
            end
        end
    end
end

```

```

elseif (VSCPCtrl(ii) == 2) && (VSCVCtrl(ii) == 2)
    %' secondary powercontrol & secondary voltage control
    with m'
    if (jj==ll)
        JAC(pp,mm)= -VSCPCAL(nn2) - ((k1*mVSC(ii,jj))^2*YR1(
            ii,jj)-YR0(ii,jj))*VM(VSCrec(ii),jj)^2;
    else
        JAC(pp,mm)=0;
    end
end
end
end
end
end

% Store information (in columns) of voltage controlled nodes
if (VSCGCtrl(ii)==1 || VSCGCtrl(ii)==3)
    if (VSCVCtrl(ii) == 1)
        JC(3*(ii-1)+1:3*(ii-1)+3,1:6*(nbb+NVSC)) = JAC(:,6*(VSCsend(ii)-1)
            +4:6*(VSCsend(ii)-1)+6)';
    elseif (VSCVCtrl(ii) == 2)
        JC(3*(ii-1)+1:3*(ii-1)+3,1:6*(nbb+NVSC)) = JAC(:,6*(VSCrec(ii)-1)
            +4:6*(VSCrec(ii)-1)+6)';
    end
end
end
end

% Update the Jacobian with the store information of voltage controlled nodes
for ii = 1 : NVSC
    if (VSCGCtrl(ii)==1 || VSCGCtrl(ii)==3)
        if (VSCVCtrl(ii) == 1)
            JAC(:,6*(VSCsend(ii)-1)+4:6*(VSCsend(ii)-1)+6) = JC(3*(ii-1)
                +1:3*(ii-1)+3,1:6*(nbb+NVSC))';
        elseif (VSCVCtrl(ii) == 2)
            JAC(:,6*(VSCrec(ii)-1)+4:6*(VSCrec(ii)-1)+6) = JC(3*(ii-1)+1:3*(
                ii-1)+3,1:6*(nbb+NVSC))';
        end
    end
end
end

% Delete the voltage magnitude and phase angle equations of the slack bus
and voltage magnitude equations corresponding to PV buses
for kk = 1 : nbb
    if (bustype(kk) == 1)
        ll = (kk-1)*6 + 1;
        for ii = ll : ll + 2
            for jj = 1 : 6*(nbb+NVSC)
                if ii == jj
                    JAC(ii,ii) = 1;
                else
                    JAC(ii,jj) = 0;
                    JAC(jj,ii) = 0;
                end
            end
        end
    end
end
end
```

```

    if (bustype(kk) == 1) | (bustype(kk) == 2) | (bustype(kk) == 4)
        ll = (kk-1)*6 + 1;
        for ii = ll + 3 : ll + 5
            for jj = 1 : 6*(nbb+NVSC)
                if ii == jj
                    JAC(ii , ii) = 1;
                else
                    JAC(ii , jj) = 0;
                    JAC(jj , ii) = 0;
                end
            end
        end
    end
end
%End of function VSCNewtonRaphsonJacobian3Ph

```

StateVariablesUpdates3Ph.m

```

function [VA,VM] = StateVariablesUpdates3Ph(nbb,D,VA,VM,NVSC,VSCrec)
for ii = 1 : nbb
    iii = (ii-1)*6 + 1;
    for jj = 1 : 3
        VA(ii , jj) = VA(ii , jj) + D(iii);
        VM(ii , jj) = VM(ii , jj) + D(iii + 3)*VM(ii , jj);
        iii = iii + 1;
    end
end
% End StateVariablesUpdates3Ph

```

VSCUpdates3Ph.m

```

function [VM,VA,BVSC0,mVSC,phiVSC] = VSCUpdates3Ph(nbb,VM,VA,NVSC,VSCsend,
    VSCrec,BVSC0,VSCVMT,mVSC,phiVSC,VSCGCtrl,VSCVCtrl,VSCtype,D)
for ii = 1 : NVSC
    for jj= 1:3
        if (VSCGCtrl(ii)==1 || VSCGCtrl(ii)==3) %Voltage control or Voltage
            Power Flow Control
            if (VSCVCtrl(ii) == 1) %On sending end
                % primary voltage modulation ratio control with m
                mVSC(ii,jj) = mVSC(ii,jj) + D(6*(VSCsend(ii)-1)+3+jj)*mVSC(ii,jj)
            );
            VM(VSCsend(ii),jj) = VSCVMT(ii);
        elseif (VSCVCtrl(ii) == 2) %On receiving end
            % secondary voltage modulation ratio control with m
            mVSC(ii,jj) = mVSC(ii,jj) + D(6*(VSCrec(ii)-1)+3+jj)*mVSC(ii,jj)
            ;
            VM(VSCrec(ii),jj) = VSCVMT(ii);
        end
    end
    phiVSC(ii,jj) = phiVSC(ii,jj) + D(6*(nbb+ii-1)+jj);
    BVSC0(ii,jj) = BVSC0(ii,jj) + D(6*(nbb+ii-1)+3+jj);
    if VSCtype(ii)==2
        %VA(ii,jj)=VA(ii,jj)+ D(6*(nbb+ii-1)+jj);
        phiVSC(ii,jj)=0;
    end
end
end
% End VSCUpdates3Ph

```

VSCLimits3Ph.m

```

function [mVSC,BVSC0,VSCGCtrl,VSCVCtrl] = VSCLimits3ph(NVSC,mVSC,mVSCHi,
    mVSCLo,BVSC0,BVSC0Lo,BVSC0Hi,VSCGCtrl,VSCVCtrl,it)
if it > 1
    for ii = 1: NVSC
        for jj = 1:3
            if (VSCGCtrl(ii)==1 || VSCGCtrl(ii)==3) %Voltage control or Voltage
                Power Flow Control
                if (mVSC(ii,jj) > mVSCHi(ii)) ;
                    mVSC(ii,jj) = mVSCHi(ii) ;
                    VSCVCtrl(ii) = VSCVCtrl(ii) + 2 ;
                    'mVSCHi enabled '
                elseif (mVSC(ii,jj) < mVSCLo(ii))
                    mVSC(ii,jj) = mVSCLo(ii);
                    VSCVCtrl(ii) = VSCVCtrl(ii) + 2 ;
                    'mVSCLo enabled '
                end
            end
        end
    end
end
% End VSCLimits3Ph

```

PQFlows3Ph.m

```

function [PQsend,PQrec,PQloss,PQbus] = PQflows3Ph(nbb,ngn,ntl,nld,genbus,
    loadbus,tlsend,tlrec,PLOAD,QLOAD,VM,VA,TLImpedInv,TLAdmit);
PQsend = zeros(ntl,3);
PQrec = zeros(ntl,3);
PQloss = zeros(ntl,3);

for ii = 1 : ntl
    Vsend = ( VM(tlsend(ii,:),:).*cos(VA(tlsend(ii,:),:)) + ...
        VM(tlsend(ii,:),:).*sin(VA(tlsend(ii,:),:))*i );
    Vrec = ( VM(tlrec(ii,:),:).*cos(VA(tlrec(ii,:),:)) + ...
        VM(tlrec(ii,:),:).*sin(VA(tlrec(ii,:),:))*i );
    for jj = 1 : 5
        if jj < 4
            PQsend(ii,jj) = Vsend(1,jj)*(conj(-TLImpedInv(jj,:,ii))...
                *(Vrec(1,:))' + conj(TLImpedInv(jj,:,ii) + ...
                0.5*TLAdmit(jj,:,ii)) *(Vsend(1,:))');
            PQrec(ii,jj) = Vrec(1,jj)*(-conj(TLImpedInv(jj,:,ii))*...
                (Vsend(1,:))' + conj(TLImpedInv(jj,:,ii) + ...
                0.5*TLAdmit(jj,:,ii)) *(Vrec(1,:))');
        elseif jj == 4
            PQsend(ii,jj) = tsend(ii);
            PQrec(ii,jj) = tlrec(ii);
        else
            PQsend(ii,jj) = tlrec(ii);
            PQrec(ii,jj) = tsend(ii);
        end
        PQloss = PQsend + PQrec;
    end
end

PQbus = zeros(1,3*nbb);
for ii = 1 : ntl
    for jj = 1:3
        PQbus(3*(tsend(ii)-1)+jj) = PQbus(3*(tsend(ii)-1)+jj) + PQsend(ii,jj);
        PQbus(3*(tlrec(ii)-1)+jj) = PQbus(3*(tlrec(ii)-1)+jj) + PQrec(ii,jj);
    end
end

% Make corrections at generator buses, where there is load, in order to get
% correct generators contributions
for ii = 1 : nld
    jj = loadbus(ii);
    for kk = 1 : ngn
        ll = genbus(kk);
        if jj == ll
            for pp = 1:3
                PQbus(3*(jj-1)+pp) = PQbus(3*(jj-1)+pp) + ( PLOAD(ii,jj) + QLOAD(ii,
                    ,jj)*i );
            end
        end
    end
end
end
% End function PQflows3Ph

```

APPENDIX B. The Power Flow Program

VSCPQFlows3Ph.m

```
function [PQbus,PQVSCsend,PQVSCrec,PQVSC0,PQVSCLoss,BVSC0] = VSCPQflows3Ph (VM
    ,VA,PQbus,NVSC,VSCsend,VSCrec,RVSC1,XVSC1,GVSC0,BVSC0,mVSC,phiVSC,VSCtype
    )

PQVSCsend(NVSC,3) = zeros;
PQVSCrec(NVSC,3) = zeros;
PQVSC0(1,3*NVSC) = zeros;
PQVSCLoss(1,3*NVSC) = zeros;
% Calculate active and reactive powers at the sending and reciving ends of
    the VSC
for ii = 1 : NVSC
    if VSCtype(ii)==1
        k1 = sqrt(3/8);
    elseif VSCtype(ii)==2
        k1 = 1;
    end
    for jj= 1:3
        YVSC1 = 1/(RVSC1(ii,jj)+i*XVSC1(ii,jj));
        VV = [(VM(VSCsend(ii),jj)*cos(VA(VSCsend(ii),jj))+i*VM(VSCsend(ii),jj)*
            sin(VA(VSCsend(ii),jj)) (VM(VSCrec(ii),jj)*cos(VA(VSCrec(ii),jj))+i*
            VM(VSCrec(ii),jj)*sin(VA(VSCrec(ii),jj)))]];
        YY = [ YVSC1 -k1*mVSC(ii,jj)*YVSC1*(cos(phiVSC(ii,jj))+i*sin(phiVSC(ii,
            jj))) ; -k1*mVSC(ii,jj)*YVSC1*(cos(phiVSC(ii,jj))-i*sin(phiVSC(ii,jj)
            )) GVSC0(ii,jj)+(YVSC1+i*BVSC0(ii,jj))*(k1*mVSC(ii,jj))^2 ];
        CC = (YY*VV. ');
        PQVSC = diag(VV)*conj(CC);
        PQbus(3*(VSCsend(ii)-1)+jj) = PQbus(3*(VSCsend(ii)-1)+jj) + PQVSC(1);
        PQbus(3*(VSCrec(ii)-1)+jj) = PQbus(3*(VSCrec(ii)-1)+jj) + PQVSC(2);
        PQVSCsend(ii,jj) = PQVSC(1);
        PQVSCrec(ii,jj) = PQVSC(2);
        PQVSC0(ii,jj) = (GVSC0(ii,jj)-i*BVSC0(ii,jj))*VM(VSCrec(ii),jj)^2;
        PQVSCLoss(3*(ii-1)+jj) = PQVSCsend(ii,jj) + PQVSCrec(ii,jj);
    end
end
% End function VSCPQflows3Ph
```

APPENDIX C. THE STATE ESTIMATION PROGRAM

SE3PH-VSC-Start.m

```

% *****
% * THREE PHASE STATE ESTIMATION WITH STATCOM *
% * MAIN PROGRAM *
% *****

clear , clc ;

Test-Microgrid ;

% Initial Calculations
[YR,YI] = YBus3Ph(nbb,ntl,tlsend,tlrec,TLImpedInv,TLAdmit,ntf,tfsend,tfrec,
    tftap,TFAdmit,nsh,shbus,shresis,shreac);
[NM,NSV,z,x,xin,R,B,k,Line,BadData] = InitialComp(VM,VA,NVx,Vn,Vx,dVx,NPx,Px,
    dPx,NQx,Qx,dQx,NPxy,Pxy,dPxy,NPyx,Pyx,dPyx,NQxy,Qxy,dQxy,NQyx,Qyx,dQyx,
    nbb,ntl,tlsend,tlrec,TLAdmit,NVSC,VSCsend,VSCrec,VSCGCtrl,VSCVCtrl,mVSC,
    phiVSC,BVSC0);

% State Estimation Calculations
while (BadData==1)
    if k<1
        'Too small number of input measurements, Redundancy error'
        break
    else
        [Hx,e,ztrue,x,InvGx,VM,VA,mVSC,phiVSC,BVSC0,it] = LSM(NM,NSV,Vn,NVx,Vx,
            Pn,NPx,Qn,NQx,NPxy,NQxy,NPyx,NQyx,nbb,bustype,ntl,Psenda,Preca,Psendb,
            Precb,Qsenda,Qreca,Qsendb,Qrecb,VM,VA,YR,YI,x,z,R,B,itmax,tol,NVSC,
            VSCsend,VSCrec,RVSC1,XVSC1,GVSC0,BVSC0,mVSC,phiVSC,VSCGCtrl,VSCVCtrl,
            VSCPCtrl,VSCPF,VSCVMT,VSCtype);
        [BadData,f,chi] = IdentificationBadData(NM,R,e,k,alpha);

% Bad data test and reorganisation
        if BadData==1
            [bad,badvalue,Rprim,SE] = BadDataValue(NM,Hx,R,e,InvGx);
            [bb,ph,flag,NVx,Vn,Vx,dVx,NPx,Pn,Px,dPx,NQx,Qn,Qx,dQx,NPxy,Psenda,Preca,
                Pxy,dPxy,NPyx,Psendb,Precb,Pyx,dPyx,NQxy,...
                Qsenda,Qreca,Qxy,dQxy,NQyx,Qsendb,Qrecb,Qyx,dQyx,k,z,ztrue,Hx,e,R,x,
                VM,VA,NM] = RearrangeData(k,NM,bad,NVx,...
                Vn,Vx,dVx,NPx,Pn,Px,dPx,NQx,Qn,Qx,dQx,NPxy,Psenda,Preca,Pxy,dPxy,
                NPyx,Psendb,Precb,Pyx,dPyx,NQxy,Qsenda,Qreca,...
                Qxy,dQxy,NQyx,Qsendb,Qrecb,Qyx,dQyx,z,ztrue,Hx,e,R,x,xin,VM,VA,nbb);
            end
        end
    end
end
end

```


APPENDIX C. The State Estimation Program

```
'Final Results (VA in Degrees)'  
it  
VM  
VA = VA*180/pi  
mVSC  
phi = phiVSC*180/pi  
BVSC0  
% End of Three Phase State Estimation Main program  
  
%      Written by Tom Rubbrecht , Tampere 2016  
%      Inspired by Positive Sequence Power System State Estimation –  
%      Grzegorz Swiszc ,  
%      Glasgow 2004
```

Test-Microgrid.m

```
%      *****  
%      *   THREE PHASE STATE ESTIMATION WITH STATCOM *  
%      *           MEASUREMENT DATA           *  
%      *****  
  
% Inport data from PF3Ph_VSC:  
load('C:\...\Power Flows 3Ph\measurements.mat');  
  
% Measurements  
alpha = 0.95; %For a 95% requirement which must be satisfied  
  
for nn=1:NVSC  
%Voltage magnitudes  
NVx=size (VM,1) ;  
for ii =1:NVx  
    Vn(ii)=ii ;  
    for jj=1:3  
        Vx(ii ,jj)= VM(ii ,jj) ;  
        dVx(ii ,jj)= 0.01; % Choose accuracy measurement  
    end  
end
```

APPENDIX C. The State Estimation Program

```
%Active powers
NPx=length(PCAL)/3;
kk=1;
for ii=1:NPx
    Pn(ii)=ii;
    for jj=1:3
        Px(ii,jj)=PCAL(kk);
        for nn=1:NVSC
            if Pn(ii)==VSCsend(nn)
                Px(ii,jj)=Px(ii,jj)+VSCPCAL(6*(nn-1)+jj);
            end
            if Pn(ii)==VSCrec(nn)
                Px(ii,jj)=Px(ii,jj)+VSCPCAL(6*(nn-1)+jj+3);
            end
        end
        dPx(ii,jj)=0.01; % Choose accuracy measurement
        kk=kk+1;
    end
end

%Reactive power
NQx=length(QCAL)/3;
kk=1;
for ii=1:NQx
    Qn(ii)=ii;
    for jj=1:3
        Qx(ii,jj)=QCAL(kk);
        for nn=1:NVSC
            if Qn(ii)==VSCsend(nn)
                Qx(ii,jj)=Qx(ii,jj)+VSCQCAL(6*(nn-1)+jj);
            end
            if Qn(ii)==VSCrec(nn)
                Qx(ii,jj)=Qx(ii,jj)+VSCQCAL(6*(nn-1)+jj+3);
            end
        end
        dQx(ii,jj)=0.01; % Choose accuracy measurement
        kk=kk+1;
    end
end

%Active power flows
NPxy=size(PQsend,1);
for ii=1:NPxy
    Psenda(ii)=real(PQsend(ii,4)); Preca(ii)=real(PQsend(ii,5));
    for jj=1:3
        Pxy(ii,jj)=real(PQsend(ii,jj));
        dPxy(ii,jj)=0.01; % Choose accuracy measurement
    end
end
```

APPENDIX C. The State Estimation Program

```
%Invers active power flows
NPyx=size(PQrec,1);
for ii=1:NPyx
    Psendb(ii)=real(PQrec(ii,5)); Precb(ii)=real(PQrec(ii,4));
    for jj=1:3
        Pyx(ii,jj)=real(PQrec(ii,jj));
        dPyx(ii,jj)=0.01; % Choose accuracy measurement
    end
end

%Reactive power flows
NQxy=size(PQsend,1);
for ii=1:NQxy
    Qsenda(ii)=real(PQsend(ii,4)); Qreca(ii)=real(PQsend(ii,5));
    for jj=1:3
        Qxy(ii,jj)=imag(PQsend(ii,jj));
        dQxy(ii,jj)=0.01; % Choose accuracy measurement
    end
end

%Invers reactive power flows
NQyx=size(PQrec,1);
for ii=1:NQyx
    Qsendb(ii)=real(PQrec(ii,5)); Qrecb(ii)=real(PQrec(ii,4));
    for jj=1:3
        Qyx(ii,jj)=imag(PQrec(ii,jj));
        dQyx(ii,jj)=0.01; % Choose accuracy measurement
    end
end

%Modulation amplitude
for jj=1:3
    mVSCmeas(nn,jj)=mVSC(nn,jj);
end
```

APPENDIX C. The State Estimation Program

```
%Control Powerflow mismatches
for jj=1:3
    if VSCPCtrl(nn)==1
        Pvr0(nn,jj)=real(PQVSCsend(nn,jj));
        dPvr0(nn,jj)=0.01;
        Qvr0(nn,jj)=real(PQVSCsend(nn,jj));
        dQvr0(nn,jj)=0.01;
    end
    if VSCPCtrl(nn)==2
        Pvr0(nn,jj)=real(PQVSCrec(nn,jj));
        dPvr0(nn,jj)=0.01;
        Qvr0(nn,jj)=real(PQVSCrec(nn,jj));
        dQvr0(nn,jj)=0.01;
    end
    if VSCPCtrl(nn)==0
        Pvr0(nn,jj)=0;
        dPvr0(nn,jj)=0.01;
        Qvr0(nn,jj)=0;
        dQvr0(nn,jj)=0.01;
    end
end
end
% Removing unnecessary import variables & Avoid confusion later on
clear VM mVSC PCAL QCAL VSCPCAL VSCQCAL PQsend PQrec PQVSCsend PQVSCrec NVSC
      VSCsend VSCrec VSCPFctrl

% Busbars data
%bustype 1 :slack bus (VM,VA)
%bustype 2 :generator or PV bus
%bustype 3 :load or PQ bus
nbb=23;
bustype(1)=1;
    VM(1,1)=1.05; VA(1,1)=0;...
    VM(1,2)=1.05; VA(1,2)=-120*pi/180;...
    VM(1,3)=1.05; VA(1,3)=120*pi/180;
bustype(2)=3;
    VM(2,1)=1.00; VA(2,1)=0;...
    VM(2,2)=1.00; VA(2,2)=-120*pi/180;...
    VM(2,3)=1.00; VA(2,3)=120*pi/180;
bustype(3)=3;
    VM(3,1)=1.00; VA(3,1)=0;...
    VM(3,2)=1.00; VA(3,2)=-120*pi/180;...
    VM(3,3)=1.00; VA(3,3)=120*pi/180;
bustype(4)=3;
    VM(4,1)=1.00; VA(4,1)=0;...
    VM(4,2)=1.00; VA(4,2)=-120*pi/180;...
    VM(4,3)=1.00; VA(4,3)=120*pi/180;
bustype(5)=3;
    VM(5,1)=1.00; VA(5,1)=0;...
    VM(5,2)=1.00; VA(5,2)=-120*pi/180;...
    VM(5,3)=1.00; VA(5,3)=120*pi/180;
```

APPENDIX C. The State Estimation Program

```
bustype(6)=3;
    VM(6,1)=1.00; VA(6,1)=0;...
    VM(6,2)=1.00; VA(6,2)=-120*pi/180;...
    VM(6,3)=1.00; VA(6,3)=120*pi/180;
bustype(7)=3;
    VM(7,1)=1.00; VA(7,1)=0;...
    VM(7,2)=1.00; VA(7,2)=-120*pi/180;...
    VM(7,3)=1.00; VA(7,3)=120*pi/180;
bustype(8)=3;
    VM(8,1)=1.00; VA(8,1)=0;...
    VM(8,2)=1.00; VA(8,2)=-120*pi/180;...
    VM(8,3)=1.00; VA(8,3)=120*pi/180;
bustype(9)=3;
    VM(9,1)=1.00; VA(9,1)=0;...
    VM(9,2)=1.00; VA(9,2)=-120*pi/180;...
    VM(9,3)=1.00; VA(9,3)=120*pi/180;
bustype(10)=3;
    VM(10,1)=1.00; VA(10,1)=0;...
    VM(10,2)=1.00; VA(10,2)=-120*pi/180;...
    VM(10,3)=1.00; VA(10,3)=120*pi/180;
bustype(11)=3;
    VM(11,1)=1.00; VA(11,1)=0;...
    VM(11,2)=1.00; VA(11,2)=-120*pi/180;...
    VM(11,3)=1.00; VA(11,3)=120*pi/180;
bustype(12)=3;
    VM(12,1)=1.00; VA(12,1)=0;...
    VM(12,2)=1.00; VA(12,2)=-120*pi/180;...
    VM(12,3)=1.00; VA(12,3)=120*pi/180;
bustype(13)=3;
    VM(13,1)=1.00; VA(13,1)=0;...
    VM(13,2)=1.00; VA(13,2)=-120*pi/180;...
    VM(13,3)=1.00; VA(13,3)=120*pi/180;
bustype(14)=3;
    VM(14,1)=1.00; VA(14,1)=0;...
    VM(14,2)=1.00; VA(14,2)=-120*pi/180;...
    VM(14,3)=1.00; VA(14,3)=120*pi/180;
bustype(15)=3;
    VM(15,1)=1.00; VA(15,1)=0;...
    VM(15,2)=1.00; VA(15,2)=-120*pi/180;...
    VM(15,3)=1.00; VA(15,3)=120*pi/180;
bustype(16)=3;
    VM(16,1)=1.00; VA(16,1)=0;...
    VM(16,2)=1.00; VA(16,2)=-120*pi/180;...
    VM(16,3)=1.00; VA(16,3)=120*pi/180;
bustype(17)=3;
    VM(17,1)=2.00; VA(17,1)=0;...
    VM(17,2)=2.00; VA(17,2)=0;...
    VM(17,3)=2.00; VA(17,3)=0;
bustype(18)=3;
    VM(18,1)=2.00; VA(18,1)=0;...
    VM(18,2)=2.00; VA(18,2)=0;...
    VM(18,3)=2.00; VA(18,3)=0;
```

APPENDIX C. The State Estimation Program

```
bustype(19)=3;
    VM(19,1)=2.00; VA(19,1)=0;...
    VM(19,2)=2.00; VA(19,2)=0;...
    VM(19,3)=2.00; VA(19,3)=0;
bustype(20)=3;
    VM(20,1)=2.00; VA(20,1)=0;...
    VM(20,2)=2.00; VA(20,2)=0;...
    VM(20,3)=2.00; VA(20,3)=0;
bustype(21)=3;
    VM(21,1)=2.00; VA(21,1)=0;...
    VM(21,2)=2.00; VA(21,2)=0;...
    VM(21,3)=2.00; VA(21,3)=0;
bustype(22)=2;
    VM(22,1)=2.00; VA(22,1)=0;...
    VM(22,2)=2.00; VA(22,2)=0;...
    VM(22,3)=2.00; VA(22,3)=0;
bustype(23)=2;
    VM(23,1)=2.00; VA(23,1)=0;...
    VM(23,2)=2.00; VA(23,2)=0;...
    VM(23,3)=2.00; VA(23,3)=0;

% Transmission lines data
ntl=6;
tlsen(1)=2; tlr(1)=3;
    lengthcable(1)=8;
tlsen(2)=2; tlr(2)=5;
    lengthcable(2)=9;
tlsen(3)=3; tlr(3)=8;
    lengthcable(3)=10;
tlsen(4)=5; tlr(4)=14;
    lengthcable(4)=11;
tlsen(5)=8; tlr(5)=11;
    lengthcable(5)=10;
tlsen(6)=11; tlr(6)=14;
    lengthcable(6)=9;

Z_nonbonded=[0.1700+0.1653*i 0.0945-0.0127*i 0.0732-0.0286*i; 0.0945-0.0127*
    i 0.1583+0.1505*i 0.0945-0.0127*i; 0.0732-0.0286*i 0.0945-0.0127*i
    0.1700+0.1653*i];
Y_reduced=10^-6*[0.0+71.99*i 0.0-3.22*i 0.0-1.51*i; 0.0-3.22*i 0.0+72.10*i
    0.0-3.22*i; 0.0-1.51*i 0.0-3.22*i 0.0+71.99*i];

% Convert to per unit
Vref=132*10^3/sqrt(3);
Sref=100*10^6;
Zref=Vref^2/Sref;
Yref=1/Zref;
```

APPENDIX C. The State Estimation Program

```
TLImpedInv = zeros(3,3,ntl);
TLAdmit = zeros(3,3,ntl);
for ii=1:ntl
    imped=Z_nonbonded*lengthcable(ii)/Zref;
    TLImpedInv(:, :, ii) = inv(imped);
    TLAdmit(:, :, ii)=Y_reduced*lengthcable(ii)/Yref;
end

% Transformers data
ntf=10;
tfsend(1)=1; tfrec(1)=2;
    tfresis(1)=0.012; tfreac(1)=0.120; tftap(1)=0.97;
tfsend(2)=3; tfrec(2)=4;
    tfresis(2)=0.030; tfreac(2)=0.300; tftap(2)=0.99;
tfsend(3)=5; tfrec(3)=6;
    tfresis(3)=0.015; tfreac(3)=0.150; tftap(3)=0.95;
tfsend(4)=5; tfrec(4)=7;
    tfresis(4)=0.030; tfreac(4)=0.300; tftap(4)=1.05;
tfsend(5)=8; tfrec(5)=9;
    tfresis(5)=0.015; tfreac(5)=0.150; tftap(5)=0.95;
tfsend(6)=8; tfrec(6)=10;
    tfresis(6)=0.030; tfreac(6)=0.300; tftap(6)=1.05;
tfsend(7)=11; tfrec(7)=12;
    tfresis(7)=0.030; tfreac(7)=0.300; tftap(7)=0.99;
tfsend(8)=11; tfrec(8)=13;
    tfresis(8)=0.030; tfreac(8)=0.300; tftap(8)=1.03;
tfsend(9)=14; tfrec(9)=15;
    tfresis(9)=0.030; tfreac(9)=0.300; tftap(9)=0.93;
tfsend(10)=14; tfrec(10)=16;
    tfresis(10)=0.030; tfreac(10)=0.300; tftap(10)=1.02;

TFAdmit=zeros(ntf,1);
for ii=1:ntf
    Z_tf=tfresis(ii)+tfreac(ii)*i;
    TFAdmit(ii)= inv(Z_tf);
end

% Generators data
ngn=3;
genbus(1)=1;
    PGEN(1,1)=0; QGEN(1,1)=0;
    PGEN(1,2)=0; QGEN(1,2)=0;
    PGEN(1,3)=0; QGEN(1,3)=0;
    QMAX(1)=9; QMIN(1)=-9;
genbus(2)=12;
    PGEN(2,1)=250/300; QGEN(2,1)=0.0;
    PGEN(2,2)=250/300; QGEN(2,2)=0.0;
    PGEN(2,3)=250/300; QGEN(2,3)=0.0;
    QMAX(2)=9; QMIN(2)=-9;
genbus(3)=23;
    PGEN(3,1)=50/300; QGEN(3,1)=0;
    PGEN(3,2)=50/300; QGEN(3,2)=0;
    PGEN(3,3)=50/300; QGEN(3,3)=0;
    QMAX(3)=9; QMIN(3)=-9;
```

APPENDIX C. The State Estimation Program

```
% Loads data
nld=4;
loadbus(1)=22;
    PLOAD(1,1)=50/300;    QLOAD(1,1)=0.0;
    PLOAD(1,2)=50/300;    QLOAD(1,2)=0.0;
    PLOAD(1,3)=50/300;    QLOAD(1,3)=0.0;
loadbus(2)=6;
    PLOAD(2,1)=0.9*230.2/300;    QLOAD(2,1)=0.9*68.9/300;
    PLOAD(2,2)=230.2/300;    QLOAD(2,2)=68.9/300;
    PLOAD(2,3)=1.1*230.2/300;    QLOAD(2,3)=1.1*68.9/300;
loadbus(3)=9;
    PLOAD(3,1)=1.1*230.2/300;    QLOAD(3,1)=1.1*68.9/300;
    PLOAD(3,2)=230.2/300;    QLOAD(3,2)=68.9/300;
    PLOAD(3,3)=0.9*230.2/300;    QLOAD(3,3)=0.9*68.9/300;
loadbus(4)=15;
    PLOAD(4,1)=0.95*100/300;    QLOAD(4,1)=0.95*50/300;
    PLOAD(4,2)=1.05*100/300;    QLOAD(4,2)=1.05*50/300;
    PLOAD(4,3)=100/300;    QLOAD(4,3)=50/300;

% Shunts data
nsh = 0 ;
shbus(1) = 0 ;
shresis(1,1) = 0 ; shreac(1,1) = 0.0 ;
shresis(1,2) = 0 ; shreac(1,2) = 0.0 ;
shresis(1,3) = 0 ; shreac(1,3) = 0.0 ;

% General parameters
itmax=50;
tol=1e-12;
nmax=6*nbb;

% Number of VSC's in network
NVSC = 7;
VSCtype=[1;1;1;1;1;2;2];

% Connectivity with network
VSCsend=[4;7;10;13;16;17;21];
VSCrec =[17;18;19;20;21;22;23];

% VSC Rating values
VSCP=[0.0;0.0;0.0;0.0;0.0;0.0;0.0]; VSCQ=[1.0;1.0;1.0;1.0;1.0;1.0;1.0];
Edc=[sqrt(2);sqrt(2);sqrt(2);sqrt(2);sqrt(2);sqrt(2);sqrt(2)];
VSCC=[sqrt(VSCP(1)^2+VSCQ(1)^2)/(Edc(1)/sqrt(2));...
    sqrt(VSCP(2)^2+VSCQ(2)^2)/(Edc(2)/sqrt(2));...
    sqrt(VSCP(3)^2+VSCQ(3)^2)/(Edc(3)/sqrt(2));...
    sqrt(VSCP(4)^2+VSCQ(4)^2)/(Edc(4)/sqrt(2));...
    sqrt(VSCP(5)^2+VSCQ(5)^2)/(Edc(5)/sqrt(2));...
    sqrt(VSCP(6)^2+VSCQ(6)^2)/(Edc(6)/sqrt(2));...
    sqrt(VSCP(7)^2+VSCQ(7)^2)/(Edc(7)/sqrt(2));];
```


APPENDIX C. The State Estimation Program

```
% VSC impedance parameters
RVSC_01=0.01*3; XVSC_01=0.10*3; GVSC_01=0.01/3; BVSC_01=0.50/3;
RVSC_02=0.01*3; XVSC_02=0.10*3; GVSC_02=0.001/3; BVSC_02=0.50/3;

RVSC1=[RVSC_01 RVSC_01 RVSC_01;...
       RVSC_01 RVSC_01 RVSC_01;...
       RVSC_01 RVSC_01 RVSC_01;...
       RVSC_01 RVSC_01 RVSC_01;...
       RVSC_01 RVSC_01 RVSC_01;...
       RVSC_02 RVSC_02 RVSC_02;...
       RVSC_02 RVSC_02 RVSC_02];
XVSC1=[XVSC_01 XVSC_01 XVSC_01;...
       XVSC_01 XVSC_01 XVSC_01;...
       XVSC_01 XVSC_01 XVSC_01;...
       XVSC_01 XVSC_01 XVSC_01;...
       XVSC_01 XVSC_01 XVSC_01;...
       XVSC_02 XVSC_02 XVSC_02;...
       XVSC_02 XVSC_02 XVSC_02];
GVSC0=[GVSC_01 GVSC_01 GVSC_01;
       GVSC_01 GVSC_01 GVSC_01;...
       GVSC_01 GVSC_01 GVSC_01;...
       GVSC_01 GVSC_01 GVSC_01;...
       GVSC_01 GVSC_01 GVSC_01;...
       GVSC_02 GVSC_02 GVSC_02;...
       GVSC_02 GVSC_02 GVSC_02];
BVSC0=[BVSC_01 BVSC_01 BVSC_01;...
       BVSC_01 BVSC_01 BVSC_01;...
       BVSC_01 BVSC_01 BVSC_01;...
       BVSC_01 BVSC_01 BVSC_01;...
       BVSC_01 BVSC_01 BVSC_01;...
       BVSC_02 BVSC_02 BVSC_02;...
       BVSC_02 BVSC_02 BVSC_02];

% VSC Initial values
mVSC= [1.0 1.0 1.0;...
       1.0 1.0 1.0;...
       1.0 1.0 1.0;...
       1.0 1.0 1.0;...
       1.0 1.0 1.0;...
       1.0 1.0 1.0;...
       1.0 1.0 1.0];
phiVSC=[0 -120*pi/180 120*pi/180;...
        0 -120*pi/180 120*pi/180;...
        0 -120*pi/180 120*pi/180;...
        0 -120*pi/180 120*pi/180;...
        0 -120*pi/180 120*pi/180;...
        0 0*pi/180 0*pi/180;...
        0 0*pi/180 0*pi/180];
```

APPENDIX C. The State Estimation Program

```
% VSC limits
mVSCHi = [5;5;5;5;5;99;99]; mVSCLo = [0.2;0.2;0.2;0.2;0.2;0.2;0.2];
BVSC0Lo= [0;0;0;0;0;0;0]; BVSC0Hi= [BVSC0(1);BVSC0(2);BVSC0(3);BVSC0(4);
    BVSC0(5);BVSC0(6);BVSC0(7)];

% VSC Control parameters
    %VSCGCtrl is the global control
        % = 0 is no control
        % = 1 is voltage control
        % = 2 is power flow control
        % = 3 is both voltage and power flow control
VSCGCtrl = [0;0;0;0;0;0;0];
    %VSCVCtrl is the voltage control; 1 = on sending end; 2= on receiving
    end
VSCVCtrl = [0;0;0;0;0;0;0]; VSCVMT = [1.05;1.05;1.05;1.05;1.05;1.05;1.05];
    %VSCPCtrl is the power flow control;1= on sending end (not working due
    to fundamental problems);2 = on receiving end
VSCPCtrl = [0;0;0;0;0;0;0]; VSCPF = [0.0;0.0;0.0;0.0;0.0;0.0;0.0];

clear RVSC_01 XVSC_01 GVSC_01 BVSC_01 RVSC_02 XVSC_02 GVSC_02 BVSC_02
% End of Data
```

APPENDIX C. The State Estimation Program

YBus3Ph.m

```
function [YR,YI] = YBus3Ph(nbb, ntl, tlsend, tlrec, TLImpedInv, TLAdmit, ntf,
    tfsend, tfrec, tftap, TFAdmit, nsh, shbus, shresis, shreac)
YY = zeros(nbb*3,nbb*3);

% Transmission lines contribution
for kk = 1 : ntl
    ii = (tlsend(kk)-1)*3 + 1;
    jj = (tlrec(kk)-1)*3 + 1;
    YY(ii:ii+2,ii:ii+2) = YY(ii:ii+2,ii:ii+2) + TLImpedInv(:, :, kk) +
        0.5*TLAdmit(:, :, kk);
    YY(ii:ii+2,jj:jj+2) = YY(ii:ii+2,jj:jj+2) - TLImpedInv(:, :, kk);
    YY(jj:jj+2,ii:ii+2) = YY(jj:jj+2,ii:ii+2) - TLImpedInv(:, :, kk);
    YY(jj:jj+2,jj:jj+2) = YY(jj:jj+2,jj:jj+2) + TLImpedInv(:, :, kk) +
        0.5*TLAdmit(:, :, kk);
end

% Transformer contribution
for kk = 1: ntf
    ii = (tfsend(kk)-1)*3 + 1;
    jj = (tfrec(kk)-1)*3 + 1;
    for pp=1:3
        YY(ii+pp-1,ii+pp-1) = YY(ii+pp-1,ii+pp-1) + TFAdmit(kk);
        YY(ii+pp-1,jj+pp-1) = YY(ii+pp-1,jj+pp-1) - tftap(kk)*TFAdmit(kk);
        YY(jj+pp-1,ii+pp-1) = YY(jj+pp-1,ii+pp-1) - tftap(kk)*TFAdmit(kk);
        YY(jj+pp-1,jj+pp-1) = YY(jj+pp-1,jj+pp-1) + tftap(kk)^2*TFAdmit(kk);
    end
end

% Shunt elements contribution
for kk = 1 : nsh
    SHAdmit = zeros(3,3);
    jj = shbus(kk)*3;
    for ii = 1 : 3
        SHAdmit(ii,ii) = 1/(shresis(kk,ii) + shreac(kk,ii)*i);
    end
    YY(jj-2:jj, jj-2:jj) = YY(jj-2:jj, jj-2:jj) + SHAdmit(:, :);
end
YR = real(YY);
YI = imag(YY);
% End YBus3Ph
```

APPENDIX C. The State Estimation Program

InitialComp.m

```
function [NM,NSV,z,x,xin,R,B,k,Line,BadData] = InitialComp(VM,VA,NVx,Vn,Vx,  
    dVx,NPx,Px,dPx,NQx,Qx,dQx,NPxy,Pxy,dPxy,NPyx,Pyx,dPyx,NQxy,Qxy,dQxy,NQyx,  
    Qyx,dQyx,nbb,ntl,tlsend,tlrec,TLAdmit,NVSC,VSCsend,VSCrec,VSCGCtrl,  
    VSCVCtrl,mVSC,phiVSC,BVSC0)  
  
BadData=1;  
  
% NM: total number of measurements  
NM=3*(NVx+NPx+NQx+NPxy+NPyx+NQxy+NQyx);  
  
% NSV: number of state variables  
NSV=6*nbb-3;  
for ii=1:NVSC  
    if VSCGCtrl(ii)>=2  
        NSV=NSV+6;  
    end  
end  
  
% k: number of degrees of freedom  
k=NM-NSV;  
  
% Line: line number  
Line=zeros(3*nbb,3*nbb);  
  
% B: line-charging susceptance matrix  
B=zeros(nbb,nbb);  
for ii=1:ntl  
    B(tlsend(ii),tlrec(ii))=(imag(TLAdmit(1,1,ii))+imag(TLAdmit(2,2,ii))+  
        imag(TLAdmit(3,3,ii)))/3;  
    Line(tlsend(ii),tlrec(ii))=ii;  
    Line(tlrec(ii),tlsend(ii))=ii;  
end  
  
% x: column matrix of initial values of state variables  
x1=zeros(NSV+3,1);  
for ii=1:nbb  
    for iii=1:3  
        x1(3*(ii-1)+iii,1)=VM(ii,iii);  
        x1(3*(ii-1)+iii+3*nbb,1)=VA(ii,iii);  
    end  
end
```

APPENDIX C. The State Estimation Program

```
for ii=1:NVSC
    for iii=1:3
        if VSCGCtrl>=2
            x1(6*(nbb+ii-1)+iii)= phiVSC(ii , iii) ;
            x1(6*(nbb+ii-1)+3*ii+iii)= BVSC0(ii , iii) ;
        end
        if VSCVCtrl(ii)==1
            x1(3*(VSCsend(ii)-1)+iii)= mVSC(ii , iii) ;
        end
        if VSCVCtrl(ii)==2
            x1(3*(VSCrec(ii)-1)+iii)= mVSC(ii , iii) ;
        end
    end
end
x=x1;
x(3*nbb+1:3*nbb+3,:)=[];
xin=x;

% z: column matrix of all measurements
% R: matrix of variances
z=zeros(NM,1) ;
R=zeros(NM,NM) ;
i=1;
if NVx>0
    for ii=1:NVx
        for iii=1:3
            z(i,1)=Vx(ii , iii) ;
            R(i,i)=dVx(ii , iii) ^2;
            i=i+1;
        end
    end
end
if NPx>0
    for ii=1:NPx
        for iii=1:3
            z(i,1)=Px(ii , iii) ;
            R(i,i)=dPx(ii , iii) ^2;
            i=i+1;
        end
    end
end
if NQx>0
    for ii=1:NQx
        for iii=1:3
            z(i,1)=Qx(ii , iii) ;
            R(i,i)=dQx(ii , iii) ^2;
            i=i+1;
        end
    end
end
end
```

APPENDIX C. The State Estimation Program

```
if NPxy>0
    for ii=1:NPxy
        for iii=1:3
            z(i,1)=Pxy(ii,iii);
            R(i,i)=dPxy(ii,iii)^2;
            i=i+1;
        end
    end
end
if NPyx>0
    for ii=1:NPyx
        for iii=1:3
            z(i,1)=Pyx(ii,iii);
            R(i,i)=dPyx(ii,iii)^2;
            i=i+1;
        end
    end
end
if NQxy>0
    for ii=1:NQxy
        for iii=1:3
            z(i,1)=Qxy(ii,iii);
            R(i,i)=dQxy(ii,iii)^2;
            i=i+1;
        end
    end
end
if NQyx>0
    for ii=1:NQyx
        for iii=1:3
            z(i,1)=Qyx(ii,iii);
            R(i,i)=dQyx(ii,iii)^2;
            i=i+1;
        end
    end
end

% End InitialComp
```

APPENDIX C. The State Estimation Program

LSM.m

```
function [Hx,e,ztrue,x,InvGx,VM,VA,mVSC,phiVSC,BVSC0,it] = LSM(NM,NSV,Vn,NVx
    ,Vx,Pn,NPx,Qn,NQx,NPxy,NQxy,NPyx,NQyx,nbb,bustype,ntl,Psenda,Preca,Psendb
    ,Precb,Qsenda,Qreca,Qsendb,Qrecb,VM,VA,YR,YI,x,z,R,B,itmax,tol,NVSC,
    VSCsend,VSCrec,RVSC1,XVSC1,GVSC0,BVSC0,mVSC,phiVSC,VSCGCtrl,VSCVCtrl,
    VSCPCtrl,VSCPF,VSCVMT,VSCtype)

flag=0;
it=0;

while (it<itmax && flag==0)
% CALCULATED POWERS
[PxCAL,QxCAL,PxyCAL,QxyCAL,PyxCAL,QyxCAL] = CalculatedPowers3Ph(nbb,ntl,
    Psenda,Preca,Psendb,Precb,Qsenda,Qsendb,Qreca,Qrecb,VM,VA,YR,YI,B);
% CALCULATED STATCOM POWERS
[VSCPCAL,VSCQCAL] = VSCCalculatedPowers3Ph(VM,VA,NVSC,VSCsend,VSCrec,RVSC1,
    XVSC1,GVSC0,BVSC0,mVSC,phiVSC,VSCtype);
% CONTROL POWER FLOW MISMATCH
[DPvr0,DQvr0] = VSCPowerMatches3Ph(NVSC,VSCPF,VSCPCAL,VSCQCAL,VSCGCtrl,
    VSCPCtrl);
% JACOBIAN FORMATION
[Hx] = HxJacobian(NM,NSV,PxCAL,QxCAL,Vn,NVx,Pn,NPx,NQx,NPxy,NQxy,NPyx,NQyx,
    nbb,ntl,Psenda,Preca,Psendb,Precb,Qsenda,Qreca,Qsendb,Qrecb,VM,VA,YR,YI,B
    );
% STATCOM JACOBIAN UPDATING
[Hx] = VSCJacobian3Ph(nbb,bustype,VM,VA,Hx,NVSC,VSCsend,VSCrec,mVSC,phiVSC,
    RVSC1,XVSC1,GVSC0,BVSC0,VSCPCAL,VSCQCAL,VSCGCtrl,VSCVCtrl,VSCPCtrl,
    VSCtype,NVx,NPx,NQx,NPxy,NQxy,NPyx,NQyx,Pn,Qn);
% ERRORS
[e,ztrue,x,xnew,Gx,InvGx,VM,VA,mVSC,phiVSC,BVSC0] = Errors(z,Hx,x,R,NSV,nbb,
    VM,VA,NM,PxCAL,QxCAL,PxyCAL,QxyCAL,PyxCAL,QyxCAL,NVx,Vn,Vx,NPx,Pn,NQx,Qn,
    NPxy,NQxy,NPyx,NQyx,VSCPCAL,VSCQCAL,NVSC,VSCsend,VSCrec,VSCGCtrl,VSCVCtrl
    ,VSCPCtrl,mVSC,phiVSC,BVSC0,DPvr0,DQvr0,VSCVMT);
%Check for convergence
if abs(xnew-x)<tol
    flag=1;
end
it=it+1;
x=xnew;
end
end
% End LSM
```

APPENDIX C. The State Estimation Program

CalculatedPowers3Ph.m

```
function [PxCAL,QxCAL,PxyCAL,QxyCAL,PyxCAL,QyxCAL] = CalculatedPowers3Ph(nbb
    , ntl , Psenda , Preca , Psendb , Precb , Qsenda , Qsendb , Qreca , Qrecb , VM,VA,YR,YI,B)

PxCAL = zeros(nbb,3);
QxCAL = zeros(nbb,3);
PxyCAL = zeros(ntl,3);
QxyCAL = zeros(ntl,3);
PyxCAL = zeros(ntl,3);
QyxCAL = zeros(ntl,3);
VV1=zeros(3*nbb,1);
YY=YR+YI*i;
for ii=1:nbb
    for jj=1:3
        iii=3*(ii-1)+jj;
        VV1(iii,1)= VM(ii,jj)*cos(VA(ii,jj))+i*VM(ii,jj)*sin(VA(ii,jj));
    end
end
II=YY*VV1;
for ii=1:3*nbb
    VV(ii,ii)=VV1(ii,1);
end
PP=VV*conj(II);
for ii=1:nbb
    for jj=1:3
        PxCAL(ii,jj)=real(PP(3*(ii-1)+jj));
        QxCAL(ii,jj)=imag(PP(3*(ii-1)+jj));
    end
end
%Pxy
for ii = 1 : length(Psenda)
    for jj = 1:3
        kk=Psenda(1,ii);
        mm=Preca(1,ii);
        PxyCAL(ii,jj) = -VM(kk,jj)^2*YR(3*(mm-1)+jj,3*(kk-1)+jj)+VM(kk,jj)*
            VM(mm,jj)*(YR(3*(mm-1)+jj,3*(kk-1)+jj)*cos(VA(kk,jj)-VA(mm,jj))+
            YI(3*(mm-1)+jj,3*(kk-1)+jj)*sin(VA(kk,jj)-VA(mm,jj)));
    end
end
%Qxy
for ii = 1 : length(Qsenda)
    for jj = 1:3
        kk=Qsenda(1,ii);
        mm=Qreca(1,ii);
        QxyCAL(ii,jj) = -VM(kk,jj)^2*(0.5*B(kk,mm)-YI(3*(mm-1)+jj,3*(kk-1)+
            jj))+VM(kk,jj)*VM(mm,jj)*(YR(3*(mm-1)+jj,3*(kk-1)+jj)*sin(VA(kk,
            jj)-VA(mm,jj))-YI(3*(mm-1)+jj,3*(kk-1)+jj)*cos(VA(kk,jj)-VA(mm,jj)
            )));
    end
end
end
```


APPENDIX C. The State Estimation Program

```
%Pyx
for ii = 1 : length(Psendb)
    for jj = 1:3
        kk=Psendb(1,ii);
        mm=Precb(1,ii);
        PyxCAL(ii,jj) = -VM(mm,jj)^2*YR(3*(mm-1)+jj,3*(kk-1)+jj)+VM(mm,jj)*
            VM(kk,jj)*(YR(3*(mm-1)+jj,3*(kk-1)+jj)*cos(VA(mm,jj)-VA(kk,jj))+
            YI(3*(mm-1)+jj,3*(kk-1)+jj)*sin(VA(mm,jj)-VA(kk,jj)));
    end
end
%Qyx
for ii = 1 : length(Qsendb)
    for jj = 1:3
        kk=Qsendb(1,ii);
        mm=Qrecb(1,ii);
        QyxCAL(ii,jj) = -VM(mm,jj)^2*(0.5*B(mm,kk)-YI(3*(kk-1)+jj,3*(mm-1)+
            jj))+VM(mm,jj)*VM(kk,jj)*(YR(3*(mm-1)+jj,3*(kk-1)+jj)*sin(VA(mm,
            jj)-VA(kk,jj))-YI(3*(mm-1)+jj,3*(kk-1)+jj)*cos(VA(mm,jj)-VA(kk,jj)
            )));
    end
end
% End CalculatedPowers3Ph
```

VSCCalculatedPowers3Ph.m

```
VSCPCAL = zeros(1,4*NVSC);
VSCQCAL = zeros(1,4*NVSC);

% Calculate VSC admittances
for ii = 1:NVSC
    if VSCtype(ii)==1
        k1 = sqrt(3/8);
    elseif VSCtype(ii)==2
        k1 = 1;
    end
    for jj=1:3
        D = RVSC1(ii,jj)^2 + XVSC1(ii,jj)^2 ;
        if D<1e-9
            YR1 = 0;
            YI1 = 0;
        else
            YR1 = RVSC1(ii,jj)/D;
            YI1 = -XVSC1(ii,jj)/D;
        end
    end

    YR0 = GVSC0(ii,jj);
    YI0 = BVSC0(ii,jj);
```

APPENDIX C. The State Estimation Program

```
% Calculate STATCOM powers
kk = 6*(ii-1)+jj;
VSCPCAL(kk) = YR1*VM(VSCsend(ii),jj)^2 - k1*mVSC(ii,jj)*VM(VSCsend(ii),
    jj)*VM(VSCrec(ii),jj)*(YR1*cos(VA(VSCsend(ii),jj)-VA(VSCrec(ii),jj)-
    phiVSC(ii,jj))+YI1*sin(VA(VSCsend(ii),jj)-VA(VSCrec(ii),jj)-phiVSC(ii
    ,jj))) ;
VSCQCAL(kk) = -YI1*VM(VSCsend(ii),jj)^2 - k1*mVSC(ii,jj)*VM(VSCsend(ii),
    jj)*VM(VSCrec(ii),jj)*(YR1*sin(VA(VSCsend(ii),jj)-VA(VSCrec(ii),jj)-
    phiVSC(ii,jj))-YI1*cos(VA(VSCsend(ii),jj)-VA(VSCrec(ii),jj)-phiVSC(ii
    ,jj))) ;
VSCPCAL(kk+3) = (YR0+YR1*(k1*mVSC(ii,jj))^2)*VM(VSCrec(ii),jj)^2 - k1*
    mVSC(ii,jj)*VM(VSCsend(ii),jj)*VM(VSCrec(ii),jj)*(YR1*cos(VA(VSCrec(
    ii),jj)-VA(VSCsend(ii),jj)+phiVSC(ii,jj))+YI1*sin(VA(VSCrec(ii),jj)-
    VA(VSCsend(ii),jj)+phiVSC(ii,jj))) ;
VSCQCAL(kk+3) = -((YI0+YI1)*(k1*mVSC(ii,jj))^2)*VM(VSCrec(ii),jj)^2 - k1
    *mVSC(ii,jj)*VM(VSCsend(ii),jj)*VM(VSCrec(ii),jj)*(YR1*sin(VA(VSCrec(
    ii),jj)-VA(VSCsend(ii),jj)+phiVSC(ii,jj))-YI1*cos(VA(VSCrec(ii),jj)-
    VA(VSCsend(ii),jj)+phiVSC(ii,jj))) ;

end
end
% End VSCCalculatedPowers3Ph
```

VSCPowerMismatches3Ph.m

```
function [DPvr0,DQvr0] = VSCPowerMismatches3Ph(NVSC,VSCPF,VSCPCAL,VSCQCAL,
    VSCGCtrl,VSCPCtrl)
DPvr0=zeros(NVSC,3);
DQvr0=zeros(NVSC,3);
for ii = 1 : NVSC
    if VSCGCtrl(ii) >= 2
        for jj= 1:3
            if VSCPCtrl(ii) == 1
                DPvr0(ii,jj) = VSCPF(ii) - VSCPCAL(6*(ii-1)+jj);
            elseif VSCPCtrl(ii) == 2
                DPvr0(ii,jj) = VSCPF(ii) + VSCPCAL(6*(ii-1)+3+jj);
            end
            DQvr0(ii,jj) = 0 + VSCQCAL(6*(ii-1)+3+jj);
        end
    end
end
% End VSCPowersMismatches3Ph
```

APPENDIX C. The State Estimation Program

HxJacobian.m

```
function [Hx] = HxJacobian(NM,NSV,PxCAL,QxCAL,Vn,NVx,Pn,NPx,NQx,NPxy,NQxy,  
    NPyx,NQyx,nbb,ntl,Psenda,Preca,Psendb,Precb,Qsenda,Qreca,Qsendb,Qrecb,VM,  
    VA,YR,YI,B)  
  
Hx = zeros(NM,NSV);  
% 1 and 2: Derive voltages to voltages and angles  
if NVx>0  
    for ii=1:NVx  
        if Vn(ii)==ii  
            for iii=1:3  
                Hx(3*(ii-1)+iii,3*(ii-1)+iii)=1;  
            end  
        elseif Vn(ii)==ii+1  
            for iii=1:3  
                Hx(3*(ii-1)+iii,3*(ii-1)+iii+3)=1;  
            end  
        elseif Vn(ii)==ii+2  
            for iii=1:3  
                Hx(3*(ii-1)+iii,3*(ii-1)+iii+6)=1;  
            end  
        end  
    end  
end  
% 3 and 4: Derive active power injections to voltages and angles  
if NPx>0  
    for ii = 1 : NPx  
        kk =(ii-1)*3 + 1;  
        for mm=1:3;  
            for jj = 1 : nbb  
                ll = (jj-1)*3 + 1;  
                for nn=1:3;  
                    if ii == jj  
                        if mm==nn  
                            Hx(3*NVx+kk+mm-1,ll+nn-1)= PxCAL(ii,mm)/VM(ii,mm)+VM(  
                                ii,mm)*YR(kk+mm-1,kk+mm-1);  
                        else  
                            Hx(3*NVx+kk+mm-1,ll+nn-1)=VM(ii,mm)*(YR(kk+mm-1,kk+nn  
                                -1)*cos(VA(ii,mm)-VA(ii,nn))+YI(kk+mm-1,kk+nn-1)*  
                                sin(VA(ii,mm)-VA(ii,nn)));  
                        end  
                    else  
                        Hx(3*NVx+kk+mm-1,ll+nn-1)=VM(ii,mm)*(YR(kk+mm-1,ll+nn-1)*  
                            cos(VA(ii,mm)-VA(jj,nn))+YI(kk+mm-1,ll+nn-1)*sin(VA(ii  
                                ,mm)-VA(jj,nn)));  
                    end  
                end  
            end  
        end  
    end
```

APPENDIX C. The State Estimation Program

```

for jj = 2 : nbb
    ll = (jj-2)*3+1;
    for nn=1:3;
        if ii == jj
            if mm==nn
                Hx(3*Nvx+kk+mm-1,3*nbb+ll+nn-1)= -QxCAL(ii,mm)-(VM(ii,mm)^2)*YI(kk+mm-1,kk+mm-1);
            else
                Hx(3*Nvx+kk+mm-1,3*nbb+ll+nn-1)=VM(ii,mm)*VM(jj,nn)*(YR(kk+mm-1,kk+nn-1)*sin(VA(ii,mm)-VA(ii,nn))-YI(kk+mm-1,kk+nn-1)*cos(VA(ii,mm)-VA(ii,nn)));
            end
        else
            Hx(3*Nvx+kk+mm-1,3*nbb+ll+nn-1)=VM(ii,mm)*VM(jj,nn)*(YR(kk+mm-1,3+ll+nn-1)*sin(VA(ii,mm)-VA(jj,nn))-YI(kk+mm-1,3+ll+nn-1)*cos(VA(ii,mm)-VA(jj,nn)));
        end
    end
end
end
end
end
% 5 and 6: Derive reactive power injections to voltages and angles
if NQx>0
    for ii = 1 : NQx
        kk =(ii-1)*3 + 1;
        jjj = 1;
        for mm=1:3;
            for jj = 1 : nbb
                ll = (jj-1)*3 + 1;
                for nn=1:3;
                    if ii == jj
                        if mm==nn
                            Hx(3*Nvx+3*NPx+kk+mm-1,ll+nn-1)= QxCAL(ii,mm)/VM(ii,mm)-VM(ii,mm)*YI(kk+mm-1,kk+mm-1);
                        else
                            Hx(3*Nvx+3*NPx+kk+mm-1,ll+nn-1)=-VM(ii,mm)*(YR(kk+mm-1,kk+nn-1)*sin(VA(ii,mm)-VA(jj,nn))+YI(kk+mm-1,kk+nn-1)*cos(VA(ii,mm)-VA(jj,nn)));
                        end
                    else
                        Hx(3*Nvx+3*NPx+kk+mm-1,ll+nn-1)=-VM(ii,mm)*(YR(kk+mm-1,ll+nn-1)*sin(VA(ii,mm)-VA(jj,nn))+YI(kk+mm-1,ll+nn-1)*cos(VA(ii,mm)-VA(jj,nn)));
                    end
                end
            end
        end
    end
end

```

APPENDIX C. The State Estimation Program

```

for jj = 2 : nbb
    ll = (jj-2)*3 + 1;
    for nn=1:3;
        if ii == jj
            if mm==nn
                Hx(3*Nvx+3*NPx+kk+mm-1,3*nbb+ll+nn-1)= PxCAL(ii,mm)-(
                    VM(ii,mm)^2)*YR(kk+mm-1,kk+mm-1);
            else
                Hx(3*Nvx+3*NPx+kk+mm-1,3*nbb+ll+nn-1)=VM(ii,mm)*VM(
                    jj,nn)*(YR(kk+mm-1,kk+nn-1)*cos(VA(ii,mm)-VA(jj,nn)
                    ))-YI(kk+mm-1,kk+nn-1)*sin(VA(ii,mm)-VA(jj,nn)));
            end
        else
            Hx(3*Nvx+3*NPx+kk+mm-1,3*nbb+ll+nn-1)=VM(ii,mm)*VM(jj,nn)
            *(YR(kk+mm-1,3+ll+nn-1)*cos(VA(ii,mm)-VA(jj,nn)))-YI(
                kk+mm-1,3+ll+nn-1)*sin(VA(ii,mm)-VA(jj,nn)));
        end
    end
end
end
end
end
end
% 7 and 8: Derive power flow to voltages and angles
if NPxy>0
    for ii = 1 : NPxy
        kk =(ii-1)*3 + 1;
        senda=Psenda(1,ii);
        reca=Preca(1,ii);
        for mm=1:3;
            for jj = 1 : nbb
                ll = (jj-1)*3 + 1;
                for nn=1:3;
                    if mm==nn
                        if senda == jj
                            Hx(3*Nvx+3*NPx+3*NQx+kk+mm-1,ll+nn-1)= -2*VM(senda,mm)
                                *YR(3*(reca-1)+mm,3*(senda-1)+mm)+VM(reca,mm)*(YR
                                    (3*(reca-1)+mm,3*(senda-1)+mm)*cos(VA(senda,mm)-VA
                                        (reca,mm))+YI(3*(reca-1)+mm,3*(senda-1)+mm)*sin(VA
                                            (senda,mm)-VA(reca,mm)));
                        end
                        if reca == jj
                            Hx(3*Nvx+3*NPx+3*NQx+kk+mm-1,ll+nn-1)= VM(senda,mm)*(
                                YR(3*(reca-1)+mm,3*(senda-1)+mm)*cos(VA(senda,mm)-
                                    VA(reca,mm))+YI(3*(reca-1)+mm,3*(senda-1)+mm)*sin(
                                        VA(senda,mm)-VA(reca,mm)));
                        end
                    end
                end
            end
        end
    end
end
end

```

APPENDIX C. The State Estimation Program

```

for jj = 2 : nbb
    ll = (jj-2)*3 + 1;
    for nn=1:3;
        if mm==nn
            if senda == jj
                Hx(3*Nvx+3*NPx+3*NQx+kk+mm-1,3*nbb+ll+nn-1)= VM(senda
                    ,mm)*VM(reca ,mm)*(-YR(3*( reca -1)+mm,3*( senda -1)+mm)
                        ) *sin(VA(senda ,mm)-VA(reca ,mm))+YI(3*( reca -1)+mm
                            ,3*( senda -1)+mm)*cos(VA(senda ,mm)-VA(reca ,mm)) );
            end
            if reca == jj
                Hx(3*Nvx+3*NPx+3*NQx+kk+mm-1,3*nbb+ll+nn-1)= VM(senda
                    ,mm)*VM(reca ,mm)*(YR(3*( reca -1)+mm,3*( senda -1)+mm)
                        ) *sin(VA(senda ,mm)-VA(reca ,mm))-YI(3*( reca -1)+mm
                            ,3*( senda -1)+mm)*cos(VA(senda ,mm)-VA(reca ,mm)) );
            end
        end
    end
end
end
end
% 9 and 10: Derive inverse power flow to voltages and angles
if NPyx>0
for ii = 1 : NPyx
    kk =(ii-1)*3 + 1;
    sendb=Psendb(1,ii);
    recb=Precb(1,ii);
    for mm=1:3;
        for jj = 1 : nbb
            ll = (jj-1)*3 + 1;
            for nn=1:3;
                if mm==nn
                    if sendb == jj
                        Hx(3*Nvx+3*NPx+3*NQx+3*NPxy+kk+mm-1,ll+nn-1)= VM(recb
                            ,mm)*(YR(3*( sendb -1)+mm,3*( recb -1)+mm)*cos(VA(recb
                                ,mm)-VA(sendb ,mm))+YI(3*( sendb -1)+mm,3*( recb -1)+mm)
                                    ) *sin(VA(recb ,mm)-VA(sendb ,mm)) );
                    end
                    if recb == jj
                        Hx(3*Nvx+3*NPx+3*NQx+3*NPxy+kk+mm-1,ll+nn-1)= -2*VM(
                            recb ,mm)*YR(3*( sendb -1)+mm,3*( recb -1)+mm)+VM(sendb
                                ,mm)*(YR(3*( sendb -1)+mm,3*( recb -1)+mm)*cos(VA(recb
                                    ,mm)-VA(sendb ,mm))+YI(3*( sendb -1)+mm,3*( recb -1)+mm)
                                        ) *sin(VA(recb ,mm)-VA(sendb ,mm)) );
                    end
                end
            end
        end
    end
end

```

APPENDIX C. The State Estimation Program

```

for jj = 2 : nbb
    ll = (jj-2)*3 + 1;
    for nn=1:3;
        if mm==nn
            if sendb == jj
                Hx(3*Nvx+3*NPx+3*NQx+3*NPxy+kk+mm-1,3*nbb+ll+nn-1)=
                    VM(sendb,mm)*VM(recb,mm)*(YR(3*(sendb-1)+mm,3*(
                        recb-1)+mm)*sin(VA(recb,mm)-VA(sendb,mm))-YI(3*(
                            sendb-1)+mm,3*(recb-1)+mm)*cos(VA(recb,mm)-VA(
                                sendb,mm))));
            end
            if recb == jj
                Hx(3*Nvx+3*NPx+3*NQx+3*NPxy+kk+mm-1,3*nbb+ll+nn-1)=
                    VM(sendb,mm)*VM(recb,mm)*(-YR(3*(sendb-1)+mm,3*(
                        recb-1)+mm)*sin(VA(recb,mm)-VA(sendb,mm))+YI(3*(
                            sendb-1)+mm,3*(recb-1)+mm)*cos(VA(recb,mm)-VA(
                                sendb,mm))));
            end
        end
    end
end
end
end
% 11 and 12: Derive reactive power flow to voltages and angles
if NQxy>0
    for ii = 1 : NQxy
        kk =(ii-1)*3 + 1;
        senda=Qsenda(1,ii);
        reca=Qreca(1,ii);
        for mm=1:3;
            for jj = 1 : nbb
                ll = (jj-1)*3 + 1;
                for nn=1:3;
                    if mm==nn
                        if senda == jj
                            Hx(3*Nvx+3*NPx+3*NQx+3*NPxy+3*NPyx+kk+mm-1,ll+nn-1)=
                                -2*VM(senda,mm)*(0.5*B(senda,reca)-YI(3*(reca-1)+
                                    mm,3*(senda-1)+mm))+VM(reca,mm)*(YR(3*(reca-1)+mm
                                        ,3*(senda-1)+mm)*sin(VA(senda,mm)-VA(reca,mm))-YI
                                            (3*(reca-1)+mm,3*(senda-1)+mm)*cos(VA(senda,mm)-VA
                                                (reca,mm))));
                            end
                            if reca == jj
                                Hx(3*Nvx+3*NPx+3*NQx+3*NPxy+3*NPyx+kk+mm-1,ll+nn-1)=
                                    VM(senda,mm)*(YR(3*(reca-1)+mm,3*(senda-1)+mm)*sin
                                        (VA(senda,mm)-VA(reca,mm))-YI(3*(reca-1)+mm,3*(
                                            senda-1)+mm)*cos(VA(senda,mm)-VA(reca,mm)));
                            end
                        end
                    end
                end
            end
        end
    end
end

```

APPENDIX C. The State Estimation Program

```

for jj = 2 : nbb
    ll = (jj-2)*3 + 1;
    for nn=1:3;
        if mm==nn
            if senda == jj
                Hx(3*Nvx+3*NPx+3*NQx+3*NPxy+3*NPyx+kk+mm-1,3*nbb+ll+
                    nn-1)= VM(senda ,mm)*VM(reca ,mm)*(YR(3*(reca-1)+mm
                        ,3*(senda-1)+mm)*cos(VA(senda ,mm)-VA(reca ,mm))+YI
                            (3*(reca-1)+mm,3*(senda-1)+mm)*sin(VA(senda ,mm)-VA
                                (reca ,mm))));
            end
            if reca == jj
                Hx(3*Nvx+3*NPx+3*NQx+3*NPxy+3*NPyx+kk+mm-1,3*nbb+ll+
                    nn-1)= VM(senda ,mm)*VM(reca ,mm)*(-YR(3*(reca-1)+mm
                        ,3*(senda-1)+mm)*cos(VA(senda ,mm)-VA(reca ,mm))-YI
                            (3*(reca-1)+mm,3*(senda-1)+mm)*sin(VA(senda ,mm)-VA
                                (reca ,mm))));
            end
        end
    end
end
end
end
% 13 and 14: Derive inverse reactive power flow to voltages and angles
if NQyx>0
    for ii = 1 : NQyx
        kk =(ii-1)*3 + 1;
        sendb=Qsendb(1,ii);
        recb=Qrecb(1,ii);
        for mm=1:3;
            for jj = 1 : nbb
                ll = (jj-1)*3 + 1;
                for nn=1:3;
                    if mm==nn
                        if sendb == jj
                            Hx(3*Nvx+3*NPx+3*NQx+3*NPxy+3*NPyx+3*NQxy+kk+mm-1,ll+
                                nn-1)= VM(recb ,mm)*(YR(3*(recb-1)+mm,3*(sendb-1)+
                                    mm)*sin(VA(recb ,mm)-VA(sendb ,mm))-YI(3*(recb-1)+mm
                                        ,3*(sendb-1)+mm)*cos(VA(recb ,mm)-VA(sendb ,mm)));
                            end
                            if recb == jj
                                Hx(3*Nvx+3*NPx+3*NQx+3*NPxy+3*NPyx+3*NQxy+kk+mm-1,ll+
                                    nn-1)= -2*VM(recb ,mm)*(0.5*B(recb ,sendb)-YI(3*(
                                        recb-1)+mm,3*(sendb-1)+mm))+VM(sendb ,mm)*(YR(3*(
                                            recb-1)+mm,3*(sendb-1)+mm)*sin(VA(recb ,mm)-VA(
                                                sendb ,mm))-YI(3*(recb-1)+mm,3*(sendb-1)+mm)*cos(VA
                                                    (recb ,mm)-VA(sendb ,mm))));
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end

```


APPENDIX C. The State Estimation Program

```

for jj = 2 : nbb
    ll = (jj-2)*3 + 1;
    for nn=1:3;
        if mm==nn
            if sendb == jj
                Hx(3*Nvx+3*Npx+3*NQx+3*Npxy+3*Npyx+3*NQxy+k*k+mm-1,3*
                    nbb+ll+nn-1)= VM(sendb,mm)*VM(recb,mm)*(-YR(3*(
                        recb-1)+mm,3*(sendb-1)+mm)*cos(VA(recb,mm)-VA(
                            sendb,mm))+YI(3*(recb-1)+mm,3*(sendb-1)+mm)*sin(VA(
                                recb,mm)-VA(sendb,mm)));
            end
            if recb == jj
                Hx(3*Nvx+3*Npx+3*NQx+3*Npxy+3*Npyx+3*NQxy+k*k+mm-1,3*
                    nbb+ll+nn-1)= VM(sendb,mm)*VM(recb,mm)*(YR(3*(recb
                        -1)+mm,3*(sendb-1)+mm)*cos(VA(recb,mm)-VA(sendb,mm)
                            )+YI(3*(recb-1)+mm,3*(sendb-1)+mm)*sin(VA(recb,mm)
                                )-VA(sendb,mm)));
            end
        end
    end
end
end
end
end
% End HxJacobian

```

VSCJacobian3Ph.m

```
function [Hx] = VSCJacobian3Ph(nbb, bustype, VM, VA, Hx, NVSC, VSCsend, VSCrec, mVSC
    , phiVSC, RVSC1, XVSC1, GVSC0, BVSC0, VSCPCAL, VSCQCAL, VSCGCtrl, VSCVCtrl,
    VSCPCtrl, VSCtype, NVx, NPx, NQx, NPxy, NQxy, NPyx, NQyx, Pn, Qn)

% VSC JACOBIAN MODIFICATION

YR0=zeros(NVSC,3); YR1=zeros(NVSC,3);
YI0=zeros(NVSC,3); YI1=zeros(NVSC,3);
cntrl=0;

for ii = 1 : NVSC
    if VSCtype(ii)==1
        k1 = sqrt(3/8);
    elseif VSCtype(ii)==2
        k1 = 1;
    end
end
```

APPENDIX C. The State Estimation Program

```
% Calculate VSC admittances
for jj=1:3
    D = RVSC1(ii,jj)^2 + XVSC1(ii,jj)^2;
    if D<1e-9
        YR1(ii,jj) = 0;
        YI1(ii,jj) = 0;
    else
        YR1(ii,jj) = RVSC1(ii,jj)/D ;
        YI1(ii,jj) = -XVSC1(ii,jj)/D ;
    end
    YR0(ii,jj) = GVSC0(ii,jj) ;
    YI0(ii,jj) = BVSC0(ii,jj) ;
end

for kk=1:NPx
    if Pn(kk)==VSCsend(ii)
        cntrl=1;
    end
end

% Calculate STATCOM Jacobian entries Pvr derived to Vvr,V0, theta_vr,
theta_0
JKK1=zeros(3,3); JKM1=zeros(3,3); JKK2=zeros(3,3); JKM2=zeros(3,3);
if (NPx>0 && cntrl==1)
    for jj=1:3
        nn1 = 4*(ii-1)+jj ;
        for ll=1:3
            if jj==ll
                JKK1(jj,ll)= (VSCPCAL(nn1) + YR1(ii,jj)*VM(VSCsend(ii),jj)^2)/VM
                    (VSCsend(ii),jj);
                JKM1(jj,ll)= (VSCPCAL(nn1) - YR1(ii,jj)*VM(VSCsend(ii),jj)^2)/VM
                    (VSCrec(ii),jj);
                JKK2(jj,ll)= -VSCQCAL(nn1) - YI1(ii,jj)*VM(VSCsend(ii),jj)^2;
                JKM2(jj,ll)= VSCQCAL(nn1) + YI1(ii,jj)*VM(VSCsend(ii),jj)^2;
            end
        end
    end
end

cntrl1=0;
for kk=1:NPx
    if Pn(kk)==VSCrec(ii)
        cntrl1=1;
    end
end
```

APPENDIX C. The State Estimation Program

```
% Calculate STATCOM Jacobian entries P0 derived to Vvr,V0, theta_vr, theta_0
JKK3=zeros(3,3); JKM3=zeros(3,3); JKK4=zeros(3,3); JKM4=zeros(3,3);
if (NPx>0 && cntrl1==1)
for jj=1:3
nn2 = 4*(ii-1)+jj+3;
for ll=1:3
if jj==ll
JKM3(jj,ll)=(VSCPCAL(nn2)-(YR0(ii,jj)+(k1*mVSC(ii,jj))^2*YR1(
ii,jj))*VM(VSCrec(ii,jj)^2)/(VM(VSCsend(ii,jj)));
JKK3(jj,ll)=(VSCPCAL(nn2)+(YR0(ii,jj)+(k1*mVSC(ii,jj))^2*YR1(
ii,jj))*VM(VSCrec(ii,jj)^2)/(VM(VSCrec(ii,jj)));
JKM4(jj,ll)=VSCQCAL(nn2)+((YI0(ii,jj)+YI1(ii,jj))*(k1*mVSC(ii
,jj))^2)*VM(VSCrec(ii,jj)^2;
JKK4(jj,ll)=-VSCQCAL(nn2)-((YI0(ii,jj)+YI1(ii,jj))*(k1*mVSC(ii
,jj))^2)*VM(VSCrec(ii,jj)^2;
end
end
end

cntrl2=0;
for kk=1:NQx
if Qn(kk)==VSCsend(ii)
cntrl2=1;
end
end

% Calculate STATCOM Jacobian entries Qvr derived to Vvr,V0, theta_vr,
theta_0
JKK5=zeros(3,3); JKM5=zeros(3,3); JKK6=zeros(3,3); JKM6=zeros(3,3);
if (NQx>0 && cntrl2==1)
for jj=1:3
nn1 = 4*(ii-1)+jj;
for ll=1:3
if jj==ll
JKK5(jj,ll)=(VSCQCAL(nn1)-YI1(ii,jj)*VM(VSCsend(ii,jj)^2)/(
VM(VSCsend(ii,jj)));
JKM5(jj,ll)=(VSCQCAL(nn1)+YI1(ii,jj)*VM(VSCsend(ii,jj)^2)/(
VM(VSCrec(ii,jj)));
JKK6(jj,ll)=VSCPCAL(nn1)-YR1(ii,jj)*VM(VSCsend(ii,jj)^2;
JKM6(jj,ll)=-VSCPCAL(nn1)+YR1(ii,jj)*VM(VSCsend(ii,jj)^2;
end
end
end

cntrl3=0;
for kk=1:NQx
if Qn(kk)==VSCrec(ii)
cntrl3=1;
end
end
```

APPENDIX C. The State Estimation Program

```
% Calculate STATCOM Jacobian entries Q0 derived to Vvr,V0, theta_vr, theta_0
JKK7=zeros(3,3); JKM7=zeros(3,3); JKK8=zeros(3,3); JKM8=zeros(3,3);
if (NQx>0 && cntrl3==1)
for jj=1:3
nn2 = 4*(ii-1)+jj+3;
for ll=1:3
if jj==ll
JKM7(jj,ll)=(VSCQCAL(nn2) + ((YI0(ii,jj)+YI1(ii,jj))*(k1*mVSC(
ii,jj))^2)*VM(VSCrec(ii,jj)^2)/(VM(VSCsend(ii,jj)));
JKK7(jj,ll)=(VSCQCAL(nn2) - ((YI0(ii,jj)+YI1(ii,jj))*(k1*mVSC(
ii,jj))^2)*VM(VSCrec(ii,jj)^2)/(VM(VSCrec(ii,jj)));
JKM8(jj,ll)=VSCPCAL(nn2) + (YR0(ii,jj)+(k1*mVSC(ii,jj))^2*YR1(
ii,jj))*VM(VSCrec(ii,jj)^2);
JKK8(jj,ll)=VSCPCAL(nn2) - (YR0(ii,jj)+(k1*mVSC(ii,jj))^2*YR1(
ii,jj))*VM(VSCrec(ii,jj)^2);
end
end
end
end

%Voltage control enabled? Replace derative V with m
if (VSCGCtrl(ii)==1 || VSCGCtrl(ii)==3)
if (VSCVCtrl(ii) == 1) % primary control
Hx(:,3*VSCsend(ii)-2:3*VSCsend(ii))= 0.0;
elseif (VSCVCtrl(ii) == 2) % secondary control
Hx(:,3*VSCrec(ii)-2:3*VSCrec(ii))= 0.0;
end
end

for jj=1:3
nn1 = 4*(ii-1)+jj;
nn2 = 4*(ii-1)+jj+3;
for ll=1:3
if (VSCVCtrl(ii)==1 && jj==ll) % primary control
if cntrl==1
JKK1(jj,ll)=(VSCPCAL(nn1) - YR1(ii,jj)*VM(VSCsend(ii,jj)
^2)/(mVSC(ii,jj)); %Pvr
end
if cntrl2==1
JKK5(jj,ll)=(VSCQCAL(nn1) + YI1(ii,jj)*VM(VSCsend(ii,jj)
^2)/(mVSC(ii,jj)); %Qvr
end
if cntrl1==1
JKM3(jj,ll)=(VSCPCAL(nn2) + ((k1*mVSC(ii,jj))^2*YR1(ii,jj)
-YR0(ii,jj))*VM(VSCrec(ii,jj)^2)/(mVSC(ii,jj)); %P0
end
if cntrl3==1
JKM7(jj,ll)=(VSCQCAL(nn2) - ((k1*mVSC(ii,jj))^2*(YI1(ii,jj)
+YI0(ii,jj))*VM(VSCrec(ii,jj)^2)/(mVSC(ii,jj)); %Q0
end
end
```

APPENDIX C. The State Estimation Program

```

elseif (VSCVCtrl(ii)==2 && jj==11) % secondary control
    if cntrl==1
        JKM1(jj,11)= (VSCPCAL(nn1) - YR1(ii,jj)*VM(VSCsend(ii),jj)
            ^2)/(mVSC(ii,jj)); %Pvr
    end
    if cntrl2==1
        JKM5(jj,11)= (VSCQCAL(nn1) + YI1(ii,jj)*VM(VSCsend(ii),jj)
            ^2)/(mVSC(ii,jj)); %Qvr
    end
    if cntrl1==1
        JKK3(jj,11) = (VSCPCAL(nn2) + ((k1*mVSC(ii,jj))^2*YR1(ii,jj)
            - YR0(ii,jj))*VM(VSCrec(ii),jj)^2)/(mVSC(ii,jj)); %P0
    end
    if cntrl3==1
        JKK7(jj,11) = (VSCQCAL(nn2) - ((k1*mVSC(ii,jj))^2*(YI1(ii,jj)
            +YI0(ii,jj))*VM(VSCrec(ii),jj)^2)/(mVSC(ii,jj)); %Q0
    end
end
end
end

% Add STATCOM contribution Pvr,P0, Qvr, Q0 to system Hx
aa=3*Nvx+3*(VSCsend(ii)-1)+1; bb=3*Nvx+3*(VSCrec(ii)-1)+1; cc=3*Nvx+3*
    NPx+3*(VSCsend(ii)-1)+1; dd=3*Nvx+3*NPx+3*(VSCrec(ii)-1)+1;
k1=3*(VSCsend(ii)-1)+1; k2=3*nbb+3*(VSCsend(ii)-2)+1;
m1=3*(VSCrec(ii)-1)+1; m2=3*nbb+3*(VSCrec(ii)-2)+1;
if cntrl==1
    Hx(aa:aa+2,k1:k1+2)= Hx(aa:aa+2,k1:k1+2)+ JKK1; Hx(aa:aa+2,m1:m1+2)= Hx(
        aa:aa+2,m1:m1+2)+ JKM1;
    Hx(aa:aa+2,k2:k2+2)= Hx(aa:aa+2,k2:k2+2)+ JKK2; Hx(aa:aa+2,m2:m2+2)= Hx(
        aa:aa+2,m2:m2+2)+ JKM2;
end
if cntrl1==1
    Hx(bb:bb+2,k1:k1+2)= Hx(bb:bb+2,k1:k1+2)+ JKM3; Hx(bb:bb+2,m1:m1+2)= Hx(
        bb:bb+2,m1:m1+2)+ JKK3;
    Hx(bb:bb+2,k2:k2+2)= Hx(bb:bb+2,k2:k2+2)+ JKM4; Hx(bb:bb+2,m2:m2+2)= Hx(
        bb:bb+2,m2:m2+2)+ JKK4;
end
if cntrl2==1
    Hx(cc:cc+2,k1:k1+2)= Hx(cc:cc+2,k1:k1+2)+ JKK5; Hx(cc:cc+2,m1:m1+2)= Hx(
        cc:cc+2,m1:m1+2)+ JKM5;
    Hx(cc:cc+2,k2:k2+2)= Hx(cc:cc+2,k2:k2+2)+ JKK6; Hx(cc:cc+2,m2:m2+2)= Hx(
        cc:cc+2,m2:m2+2)+ JKM6;
end
if cntrl3==1
    Hx(dd:dd+2,k1:k1+2)= Hx(dd:dd+2,k1:k1+2)+ JKM7; Hx(dd:dd+2,m1:m1+2)= Hx(
        dd:dd+2,m1:m1+2)+ JKK7;
    Hx(dd:dd+2,k2:k2+2)= Hx(dd:dd+2,k2:k2+2)+ JKM8; Hx(dd:dd+2,m2:m2+2)= Hx(
        dd:dd+2,m2:m2+2)+ JKK8;
end
end

```

APPENDIX C. The State Estimation Program

```
% POWER FLOW REGULATION
if VSCGCtrl>=2
    if VSCtype(ii)==1
        k1 = sqrt(3/8);
    elseif VSCtype(ii)==2
        k1 = 1;
    end
% Calculate STATCOM Jacobian entries Pvr derived to phi_rho
JKK1=zeros(3,3);
if (NPx>0 && cntrl==1)
for jj=1:3
    nn1 = 4*(ii-1)+jj;
    for ll=1:3
        if jj==ll
            JKK1(jj,ll)= VSCQCAL(nn1) + YI1(ii,jj)*VM(VSCsend(ii),jj)^2;
        end
    end
end
end
end

% Calculate STATCOM Jacobian entries P0 derived to phi_rho
JKK2=zeros(3,3);
if (NPx>0 && cntrl1==1)
for jj=1:3
    nn2 = 4*(ii-1)+jj+3;
    for ll=1:3
        if jj==ll
            JKK2(jj,ll)=VSCQCAL(nn2) - ((YI0(ii,jj)+YI1(ii,jj))*(k1*mVSC(ii,
            jj))^2)*VM(VSCrec(ii),jj)^2;
        end
    end
end
end
end

% Calculate STATCOM Jacobian entries Qvr derived to phi_rho
JKK3=zeros(3,3);
if (NQx>0 && cntrl2==1)
for jj=1:3
    nn1 = 4*(ii-1)+jj;
    for ll=1:3
        if jj==ll
            JKK3(jj,ll)=VSCPCAL(nn1) + YR1(ii,jj)*VM(VSCsend(ii),jj)^2;
        end
    end
end
end
end
end
```

APPENDIX C. The State Estimation Program

```
% Calculate STATCOM Jacobian entries Q0 derived to phi_rho, Beq
JKK4=zeros(3,3); JKM4=zeros(3,3);
if (NQx>0 && cntrl3==1)
for jj=1:3
nn2 = 4*(ii-1)+jj+3;
for ll=1:3
if jj==ll
JKK4(jj,ll)= VSCPCAL(nn2) - (YR0(ii,jj)+YR1(ii,jj)*(k1*mVSC(ii,
jj))^2)*VM(VSCrec(ii,jj))^2;
JKM4(jj,ll)=-(k1*mVSC(ii,ll))^2*VM(VSCrec(ii,jj))^2;
end
end
end

% Add STATCOM contribution Pvr,P0, Qvr, Q0 to system Hx
aa=3*Nvx+3*(VSCsend(ii)-1)+1; bb=3*Nvx+3*(VSCrec(ii)-1)+1; cc=3*Nvx+3*
NPx+3*(VSCsend(ii)-1)+1; dd=3*Nvx+3*NPx+3*(VSCrec(ii)-1)+1;
k1=3*nbb+3*(nbb-1)+6*(NVSC-1)+1; m1=3*nbb+3*(nbb-1)+6*(NVSC-1)+4;
if cntrl==1
Hx(aa:aa+2,k1:k1+2)= Hx(aa:aa+2,k1:k1+2)+ JKK1;
end
if cntrl1==1
Hx(bb:bb+2,k1:k1+2)= Hx(bb:bb+2,k1:k1+2)+ JKK2;
end
if cntrl2==1
Hx(cc:cc+2,k1:k1+2)= Hx(cc:cc+2,k1:k1+2)+ JKK3;
end
if cntrl3==1
Hx(dd:dd+2,k1:k1+2)= Hx(dd:dd+2,k1:k1+2)+ JKK4; Hx(dd:dd+2,m1:m1+2)= Hx(
dd:dd+2,m1:m1+2)+ JKM4;
end
end
end
```

APPENDIX C. The State Estimation Program

Errors.m

```
function [e, ztrue, x, xnew, Gx, InvGx, VM, VA, mVSC, phiVSC, BVSC0] = Errors(z, Hx, x, R,
    , NSV, nbb, VM, VA, NM, PxCAL, QxCAL, PxyCAL, QxyCAL, PyxCAL, QyxCAL, NVx, Vn, Vx, NPx,
    Pn, NQx, Qn, NPxy, NQxy, NPyx, NQyx, VSCPCAL, VSCQCAL, NVSC, VSCsend, VSCrec,
    VSCGCtrl, VSCVCtrl, VSCPCtrl, mVSC, phiVSC, BVSC0, DPvr0, DQvr0, VSCVMT)

% Calculation of the True Values
ztrue=zeros(NM,1);
Nprev=0;
if NVx>0
    for ii=1:NVx
        for jj=1:3
            ztrue(3*(ii-1)+jj,1)=VM(ii,jj);
        end
    end
    Nprev=Nprev+3*NVx;
end
if NPx>0
    for ii=1:NPx
        for jj=1:3
            ztrue(Nprev+3*(ii-1)+jj,1)=PxCAL(ii,jj);
        end
        for nn=1:NVSC
            if Pn(ii)==VSCsend(nn)
                ztrue(Nprev+3*(ii-1)+jj,1)=ztrue(Nprev+3*(ii-1)+jj,1)+VSCPCAL(6*(nn-1)+jj);
            end
            if Pn(ii)==VSCrec(nn)
                ztrue(Nprev+3*(ii-1)+jj,1)=ztrue(Nprev+3*(ii-1)+jj,1)+VSCPCAL(6*(nn-1)+3+jj);
            end
        end
    end
    Nprev=Nprev+3*NPx;
end
if NQx>0
    for ii=1:NQx
        for jj=1:3
            ztrue(Nprev+3*(ii-1)+jj,1)=QxCAL(ii,jj);
        end
        for nn=1:NVSC
            if Qn(ii)==VSCsend(nn)
                ztrue(Nprev+3*(ii-1)+jj,1)=ztrue(Nprev+3*(ii-1)+jj,1)+VSCQCAL(6*(nn-1)+jj);
            end
            if Qn(ii)==VSCrec(nn)
                ztrue(Nprev+3*(ii-1)+jj,1)=ztrue(Nprev+3*(ii-1)+jj,1)+VSCQCAL(6*(nn-1)+3+jj);
            end
        end
    end
    Nprev=Nprev+3*NQx;
end
```


APPENDIX C. The State Estimation Program

```
if NPxy>0
    for ii=1:NPxy
        for jj=1:3
            ztrue(Nprev+3*(ii-1)+jj,1)=PxyCAL(ii,jj);
        end
    end
    Nprev=Nprev+3*NPxy;
end
if NPyx>0
    for ii=1:NPyx
        for jj=1:3
            ztrue(Nprev+3*(ii-1)+jj,1)=PyxCAL(ii,jj);
        end
    end
    Nprev=Nprev+3*NPyx;
end
if NQxy>0
    for ii=1:NQxy
        for jj=1:3
            ztrue(Nprev+3*(ii-1)+jj,1)=QxyCAL(ii,jj);
        end
    end
    Nprev=Nprev+3*NQxy;
end
if NQyx>0
    for ii=1:NQyx
        for jj=1:3
            ztrue(Nprev+3*(ii-1)+jj,1)=QyxCAL(ii,jj);
        end
    end
    Nprev=Nprev+3*NQyx;
end

% Calculate the error
e=z-ztrue;
Gx=Hx'*inv(R)*Hx;
Gx=Gx+eye(length(Gx))*1e-10;
% Invert matrix Gx using Singular Value Decomposition optimization methode
[U,S,V]=svd(Gx);
for i=1:NSV
    S(i,i)=1/S(i,i);
end
InvGx=V*S*U';
```

APPENDIX C. The State Estimation Program

```
% Mismatch of estimated state variables
xnew=x+InvGx*Hx'*inv(R)*e;
% Update state variables

for ii=1:nbb
    for jj=1:3
        VM(ii,jj)= xnew(3*(ii-1)+jj,1);
        for nn=1:NVSC
            if (VSCVCtrl(nn)==1 && ii==VSCsend(nn))
                mVSC(nn,jj)= xnew(3*(ii-1)+jj,1);
                VM(VSCsend(nn),jj) = VSCVMT(nn);
            elseif VSCVCtrl(nn)==2 && ii==VSCrec(nn)
                mVSC(nn,jj)= xnew(3*(ii-1)+jj,1);
                VM(VSCrec(nn),jj) = VSCVMT(nn);
            end
        end
    end
end
iii=3*Nvx+1;
for ii=2:nbb
    for jj=1:3
        VA(ii,jj)= xnew(iii,1);
        iii=iii+1;
        for nn=1:NVSC
            if (ii==VSCrec(nn)&& VSCPCtrl(nn)==2)
                VA(ii,jj)=0;
            end
        end
    end
end
for nn=1:NVSC
    if VSCGCtrl(nn)>=2
        for jj=1:3
            phiVSC(nn,jj)= xnew(iii+6*(nn-1),1);
            iii=iii+1;
        end
        for jj=1:3
            BVSC0(nn,jj)= xnew(iii+6*(nn-1),1);
            iii=iii+1;
        end
    end
end
end
end
% End Errors
```

IdentificationBadData.m

```

function [BadData,f,chi] = IdentificationBadData(NM,R,e,k,alpha)
% Calculation of the weighted sum of square errors
f=0;
BadData=0;
for ii=1:NM
    f=f+e(ii,1)^2/R(ii,ii);
end

% Find corresponding chi-square value
chi=chi2inv(alpha,k);

% Test for bad data
if f>=chi
    BadData=1;
end
% End IdentificationBadData

```

BadDataValue.m

```

function [bad,badvalue,Rprim,SE] = BadDataValue(NM,Hx,R,e,InvGx)
% Diagonal elements of the co-variance matrix Rrim
Rprim=R-Hx*InvGx*Hx';

% Compute standarised errors SE
SE=zeros(NM,1);
for ii=1:NM
    SE(ii,1)=e(ii,1)/sqrt(Rprim(ii,ii));
end

% Find the largest error
bad=1;
badvalue=SE(1,1);
if NM>1
    for ii=2:NM
        if badvalue^2<(SE(ii,1))^2
            bad=ii;
            badvalue=SE(ii,1);
        end
    end
end
% End BadDataValue

```

RearrangeData.m

```

function [bb, ph, flag, NVx, Vn, Vx, dVx, NPx, Pn, Px, dPx, NQx, Qn, Qx, dQx, NPxy, Psenda,
    Preca, Pxy, dPxy, NPyx, Psendb, Precb, Pyx, dPyx, NQxy, Qsenda, Qreca, Qxy, dQxy, NQyx,
    Qsendb, Qrecb, Qyx, dQyx, k, z, ztrue, Hx, e, R, x, VM, VA, NM] = RearrangeData(k, NM,
    bad, NVx, Vn, Vx, dVx, NPx, Pn, Px, dPx, NQx, Qn, Qx, dQx, NPxy, Psenda, Preca, Pxy, dPxy,
    NPyx, Psendb, Precb, Pyx, dPyx, NQxy, Qsenda, Qreca, Qxy, dQxy, NQyx, Qsendb, Qrecb,
    Qyx, dQyx, z, ztrue, Hx, e, R, x, xin, VM, VA, nbb)

% Remove group of measurements with bad data, rearrange and compute once
% more with no bad data

% Decrease degrees of freedom
k=k-3;
NM=NM-3;
flag=0;
ph=0;

% Bad data in Voltage magnitudes?
badex=bad;
if badex <= 3*NVx
    bb=1+(badex-rem(badex,3))/3;
    ph=rem(badex,3);
    if ph==0
        bb=bb-1;
        ph=3;
    end
    'Bad input data - Vx', bb, ph, Vx(bb, ph)
% Delete bad data
Vn(bb) = [];
Vx(bb, :) = [];
dVx(bb, :) = [];
NVx=NVx-1;
flag=1;
end

% Bad data in Active Power injections?
badex=bad-3*NVx;
if (flag==0 && badex <= 3*NPx)
    bb=1+(badex-rem(badex,3))/3;
    ph=rem(badex,3);
    if ph==0
        bb=bb-1;
        ph=3;
    end
    'Bad input data - Px', bb, ph, Px(bb, ph)
% Delete bad data
Pn(bb) = [];
Px(bb, :) = [];
dPx(bb, :) = [];
NPx=NPx-1;
flag=1;
end

```

```

% Bad data in Reactive Power injections?
badex=badex-3*NPx;
if (flag==0 && badex<=3*NQx)
    bb=1+(badex-rem(badex,3))/3;
    ph=rem(badex,3);
    if ph==0
        bb=bb-1;
        ph=3;
    end
    'Bad input data - Qx',bb,ph,Qx(bb,ph)
% Delete bad data
Qn(bb)=[];
Qx(bb,:)=[];
dQx(bb,:)=[];
NQx=NQx-1;
flag=1;
end

% Bad data in Active Power flows?
badex=badex-3*NQx;
if (flag==0 && badex<=3*NPxy)
    bb=1+(badex-rem(badex,3))/3;
    ph=rem(badex,3);
    if ph==0
        bb=bb-1;
        ph=3;
    end
    'Bad input data - Pxy',bb,ph,Pxy(bb,ph)
% Delete bad data
Psenda(bb)=[];
Preca(bb)=[];
Pxy(bb,:)=[];
dPxy(bb,:)=[];
NPxy=NPxy-1;
flag=1;
end

% Bad data in invers Active Power flows?
badex=badex-3*NPxy;
if (flag==0 && badex<=3*NPyx)
    bb=1+(badex-rem(badex,3))/3;
    ph=rem(badex,3);
    if ph==0
        bb=bb-1;
        ph=3;
    end
    'Bad input data - Pyx',bb,ph,Pyx(bb,ph)
% Delete bad data
Psendb(bb)=[];
Precb(bb)=[];
Pyx(bb,:)=[];
dPyx(bb,:)=[];
NPyx=NPyx-1;
flag=1;
end

```

```

% Bad data in Reactive Power flows?
badex=badex-3*NPyx;
if (flag==0 && badex<=3*NQxy)
    bb=1+(badex-rem(badex,3))/3;
    ph=rem(badex,3);
    if ph==0
        bb=bb-1;
        ph=3;
    end
    'Bad input data - Qxy',bb,ph,Qxy(bb,ph)
% Delete bad data
Qsenda(bb)=[];
Qreca(bb)=[];
Qxy(bb,:)=[];
dQxy(bb,:)=[];
NQxy=NQxy-1;
flag=1;
end

% Bad data in invers Reactive Power flows?
badex=badex-3*NQyx;
if (flag==0 && badex<=3*NQyx)
    bb=1+(badex-rem(badex,3))/3;
    ph=rem(badex,3);
    if ph==0
        bb=bb-1;
        ph=3;
    end
    'Bad input data - Qyx',bb,ph,Qyx(bb,ph)
% Delete bad data
Qsendb(bb)=[];
Qrecb(bb)=[];
Qyx(bb,:)=[];
dQyx(bb,:)=[];
NQyx=NQyx-1;
flag=1;
end

% Delete bad rows and columns
if ph==1
    z(bad:bad+2,:)=[];
    ztrue(bad:bad+2,:)=[];
    R(bad:bad+2,:)=[];
    R(:,bad:bad+2)=[];
end
if ph==2
    z(bad-1:bad+1,:)=[];
    ztrue(bad-1:bad+1,:)=[];
    R(bad-1:bad+1,:)=[];
    R(:,bad-1:bad+1)=[];
end
if ph==3
    z(bad-2:bad,:)=[];
    ztrue(bad-2:bad,:)=[];
    R(bad-2:bad,:)=[];
    R(:,bad-2:bad)=[];
end
x=xin;
% End RearrangeData

```